



LAUREA
UNIVERSITY OF APPLIED SCIENCES
Together we are stronger

Raising Awareness of Gender Equality Gap: The Development of a Facebook Application for UN Women

Trang, Nguyen Thi Mai

2015 Leppävaara

Laurea University of Applied Sciences
Laurea Leppävaara

Raising Awareness of Gender Equality Gap:
The Development of a Facebook Application for UN Women

Trang, Nguyen Thi Mai
Degree Program in BIT
Bachelor's Thesis
20 June, 2014

Trang, Nguyen Thi Mai

Raising Awareness of Gender Equality Gap: The Development of a Facebook Application

Year 2015 Pages 40

Twenty years ago, the world leaders promised to close the gender equality gap at the World Conference held in Beijing. Known as the 'Beijing Promise', this promise was to assure policies promoting gender equality and empowerment would be implemented. Although little has been done to enact these policies since then, now an organization, UN Women, are attempting to deliver on this promise and fulfill the common wish among women of erasing inequalities. UN Women have an idea of a campaign that can gather people from all over the world to contribute to a world of equality.

Determined women came up with an idea of the web application "Close the gap". The application is generated in order to inspire women, men, partners, communities and businesses to participate by contributing their small actions to make the world a better place. The application is planned by using the popular social media platform Facebook. The objective of the application is to encourage audiences across the world to participate in the campaign and push for equality and rights for women and girls everywhere. The application allows people to devote their social media profile images, follow the number of people who have already contributed, like and share it on Facebook or Twitter.

After three months of the launch, the application can be considered to be successful as the number of contributors has reached approximately 8600, and that number is expected to increase to the target of 20,000 in the near future.

Table of Contents

1	Introduction	6
1.1	Objective	6
1.2	Motivation	6
1.3	Thesis statement	6
2	Literature review	7
2.1	Overview and relevant information	7
2.2	Development method - Software Development Life Cycle.....	8
3	Customer requirements.....	9
3.1	Social media campaign in general	10
3.2	Campaign insights	10
3.3	Campaign concept - Empowering Women, Empowering Humanity, Picture it!	11
3.4	Initial idea - Close the gap.....	12
4	Technical solutions	13
4.1	Facebook Login SDK.....	14
4.2	Twitter API	14
4.3	MySQL	14
4.4	Languages switching	16
4.5	jQuery.....	16
4.6	Auto page loader	16
5	Implementation	17
5.1	Frontend	17
5.1.1	HTML5 / CSS3	17
5.1.1.1	HTML5.....	18
5.1.1.2	CSS3	19
5.1.2	jQuery.....	20
5.2	Backend	21
5.2.1	Database Design	22
5.2.2	Database connection.....	24
5.2.3	Facebook SDK	24
5.2.4	Twitter SDK	26
5.2.5	Fetching data	26
5.2.6	Languages	27
6	Optimizing web performance	28
6.1	Front-end optimization	28
6.2	Back-end optimization.....	29
6.3	Google Analytics	30
7	Final result.....	30
8	Project Work and Learning Reflection	31

9	Conclusions	32
	References.....	33
	List of figures.....	34
	Appendix A : HTML Structure	35
	Appendix B: Javascript functions	36
	Appendix C: PHP functions.....	37
	Appendix D: Facebook / Twitter APIs	39

1 Introduction

UN Women is a union organization, working to ensure gender equality and protect the empowerment of women. On May 22, 2014 UN Women launched a major campaign in the memory of the 20th anniversary Fourth World Conference on Women in Beijing. The campaign mainly focuses on the gaps in gender equality and female empowerment as a result of the adoption of the 1995 Beijing Declaration and Platform for Action. An interactive application was released to encourage UN Women members to devote their profile pictures filling up the gap in gender equality.

1.1 Objective

The “Close the gap” application is developed according to the needs of the customers who are the leading members of UN Women in Finland. The purpose is to allow a user to choose what social media platform, Facebook or Twitter, to login. Once they have logged in to their account, their published information containing names, profile pictures, location, and time will be stored in the database. The user can see immediately see a popup message that shows the order number they hold, afterwards their profile pictures will be shown up between the gap.

In addition, the application expresses the total number of users who have already contributed to the campaign. The target number of users that UN Women aim to is 20,000. The user can see how many percentages of previous users have taken part in the campaign. Moreover, users are also able to share the application on Facebook or Twitter in different languages, including English, Spanish and French.

1.2 Motivation

The reason why this application was chosen as the thesis topic is because it has specific target customers, good planning and a good idea of closing gender equality. The leading members of UN Women Organization in Finland planned for this application four months before its release with a detailed explanation and a desired design. Time is saved during the development process because customers have a clear and sticky plan. In addition to the detailed plan, through the developing phase, different areas of know-how supporting the author’s expertise, including both frontend and backend development, can be learnt. The combination of knowledge about databases, such as PHP, HTML5, CSS3 and JavaScript makes the author more experienced with working with web application.

1.3 Thesis statement

The objective of this paper is to describe the process of a social media application development. The observation will include customer requirements, declaration to frontend development with examples, backend explanation and coding, and finally clarifying optimizing techniques. The information will be covered in an interesting and intelligible way that the readers can easily understand.

2 Literature review

2.1 Overview and relevant information

In order to perform the Beijing promise made twenty years ago about ensuring gender equality and closing the distance between women and men, today UN Women decide to launch a campaign under the title “Empowering women, empowering humanity, picture it !” (Un women documentation 2014, 107). With the aim raising the awareness of people about gender equality, the campaign is not only taken out under the form of television ads, it is also pushing the concept into a new level using a digital platform. Close the gap application is the result of combining campaign initial idea with social media and technology to spread the campaign over the Internet and let people in all over the world acknowledge about the importance of gender equality.

Through this thesis paper, the reader will be driven over every step necessary to build close the gap application, from sketching out initial idea into real project development. Information collected during the project includes UN Women campaign documents and technical papers as well as online sources related to technical development. Previous study of the writer personally is also helpful in this situation, specially the web-related and project management knowledge applied subtly in a combination with sources from the Internet in order to interpret the problem in an easy-assessable way for both technical and non-technical readers.

Information collected can be divided into three main categories. The first is the customer requirement documents narrate the history of Beijing Promise twenty years ago, initialize the concept and idea of the campaign and point out the requirements that UN Women desire to include in this campaign. The second is the front-end development resources support the implementation of coding the appearance of the application. These resources are about the latest web technologies such as HTML5, CSS3, and jQuery animations used in the application. And the final is the back-end development resources include information about database connection, working with Facebook SDK and Twitter APIs in login and getting user data. There are several online articles working as the instructions for making the back-end development much easier.

2.2 Development method - Software Development Life Cycle

There are several approaches used during the development of the application. Software Development Life Cycle (SDLC) used in this paper is one of the software development models that take responsible for ensuring the successful development of the software. SDLC (figure 1) consists of six phases including requirement and analysis, design, implementing, testing, deployment and maintenance (Msainion 2012).

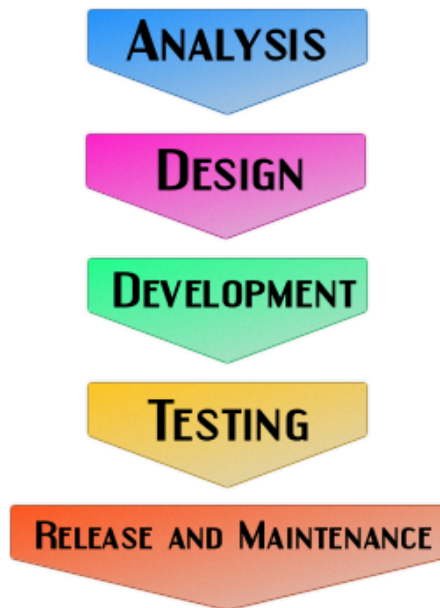


Figure 1: Software Development Life Cycle

- Requirement and analysis: Objective of this phase is collecting requirements from customers, giving observations and preparing to transfer the requirements into design mockups.
- Design: In this phase, the mockups are produced based on analyzing customer requirements. There might be several fixes during the communication between customers and designer such as colors, logos, or icons. However, customers will be responsible for determining final layout.
- Implementing (coding): Once the decided version of mockup is produced, the layout will be implemented into web application by a developer. Implementation comprises of two parts which are front-end and back-end development. Front-end coding takes over the appearance of the application and how easy the user find when using it. Vice versa, back-end coding focuses on how the system is performed and how the information is processed. Social network APIs and database connection are realized during this phase.
- Testing: This phase is considered as one of the most important elements in the development process. The application will be customized, back-end as well as front-

end errors are also debugged in this phase in order to optimize the application for better performance.

- Deployment: During this phase, the application will be checked under different kinds of environment such as mobile devices, tablets, and computers as well as Chrome, Firefox, Internet Explorer, etc.
- Maintenance: Final step in the SDLC is maintaining the application after publishing it. Any updates or extra requirements are performed during the maintenance period to create the best user experience.

In conclusion, the literature review briefs the searching process used during the project and provides the reader a better perspective about the project in general level. The rest of the paper will show in more detail of how the project is implemented and how research methodology supports the overall success of the project.

3 Customer requirements

Customer requirements explain why the application is needed and defining what it should be like. They are written from the point of view of customers and are described in narrative form. Besides, the document should explain how importance the application is and what the final version of application will be like. During the development phase, customer requirements should not be changed because it will affect the whole process. For that reason, customer requirements should be consistency from the beginning of development.

A firm requirements document behaves as guidance during the development and testing in the project. It assists when scheduling time and resources and when planning the project. Requirements should be collected, classified and then prioritized to maximize the effectiveness of the project. It is important to identify target users because requirements should come from end users. Interviewing customers to figure out what are requirements and priorities they consider, asking their opinions about which features that the application should perform, how they use them and which other features they think that might not be necessary.

Recording customer requirements is another vital step in initial phase. There might be cases that customer requirements keep changing and conflict with end user requirements. For that reason, all requirements should be documented carefully in a proper form including title, detailed description of requirements, and sources of requirements.

Once the customer requirements are recorded, they should be classified based on priority of information. Classification makes it easier to understand which part of project has high level of priority and which part has the most complicated implementation. For high priority

issues, they should be solved earliest and spent much time on to ensure that customers will be satisfied with. For complex problems, they also require a lot of time to deal with. If they do not belong to high priority listing, they can be considered later after the other important issues are solved (Un Women documentation 2014).

3.1 Social media campaign in general

The goal of social media campaign is to generate interest and awareness of the importance of the Beijing Conference Agenda. It has the role to guide, drive women's equality and empowerment effort globally, and reignite discussion and action around the key themes. The key elements of social media campaign are simple spearhead message and activation, creative content, participation options, smooth use of social media tools, cross-media, E-fluentials, coordinated launch to create sharing, management and monitoring, response and revision (Un Women 2014).

The primary target audiences of this campaign are social media users globally (a younger, under 40-age group, both men and women), online women communities globally, online media including bloggers and other influencers. The second audiences consist of print media, technology sector corporate partners, women's equality stakeholders, policy and decision-makers, senior level non-government professionals.

The social media campaign's objective is firstly to generate understanding of the importance and relevance of Beijing Conference issues today, which are about women's equality and participation in all spheres of life. Secondly, the campaign is also aim on engaging the target audiences in discussion, triggering continued long-term action in support of Beijing agenda. Thirdly, using the campaign to reach and engage online media, donors as well as key stakeholders of the post 2015 development agenda setting to ensure gender and women's issues is at the core of goals. Finally, increasing overall recognition of UN Women's role and brand globally is one of another key target of the campaign.

Some of the challenges that the campaign might face are whether the campaign is relevant to the audiences' world enough to interest them, whether it is so accessible that all audiences could understand, and what are the rewards when the audiences take part in the campaign. In order to come over those challenges, let us take a look at the campaign in more detail.

3.2 Campaign insights

Campaign insight #1: Beijing Platform for Action presents 12 big themes which can be viewed from different angles, but seen from target audiences point of view to deal one core problem, that is lack of gender equality and rights globally.

Campaign insight #2: For some people, the problem is up close but they feel they cannot do anything about it. For others, they feel too distant to act upon. The majority of people are looking for the others to solve problem (Un women 2014).

The campaign aims to inspire women, men, partners and companies to participate, the concept is scalable and universal. It is about enabling many small actions that we all can do, imagining a better world and making that a reality. By leading the way and sharing these action across the world, each of person can push for equality and rights for women and girls everywhere. By relating to the target groups everyday lives, every people can bring the Beijing Conference a message to life, and enable the others to participate, be inspired, come together to move the agenda forward.

In summary, the concept of social media campaign needs to be relatable, easy to get and create an emotional connection with the target group following three slogans “Picture it”, “Positive”, and “Inclusive”. Expected outcomes finalized in the design phase are able to engage 100,000 people in campaign material within three months of launch, increase Facebook page likes up to 200,000 within same period. In addition, the campaign also aim to promote significant online media and bloggers of the campaign and push the number of UN Women website visitors during the campaign period. Last but not least, the campaign contributes to rise the awareness of UN Women’s role in advancing Beijing agenda and positive feedback from key donor country Ministerial and Parliamentarian groups.

3.3 Campaign concept - Empowering Women, Empowering Humanity, Picture it!

UN Women has planned for a campaign a long time ago leading up to the 20th anniversary of Beijing Platform for Action as a vehicle in support of the organization’s overall strategy and its objectives for social mobilization and awareness-raising. Gradually building over the course of 2014, the campaign runs under the title “Empowering women, empowering humanity, picture it!”, and engages in Beijing’s overall theme of women’s empowerment and gender equality.

The overarching title informs a forward-looking development blueprint which stresses out the importance of ensuring women’s rights, empowerment and gender equality. “Picture it!” is an invitation message to engage in the conversation, in order to imagine a better world where gender equality and women’s empowerment is a reality and discuss the existing gaps. It is also a call-to-action, with a strong focus on visuals and imagination and continues to affect both women and men.

Because the target audiences of this campaign are global and diverse, Beijing Platform for Action will be positioned as a relevant roadmap towards achieving gender equality. In order to reach the target, the campaign should have a dynamic public engagement component with contemporized content, powerful audio-visual and social media elements that attract new audiences as well as the media.

The campaign will provide opportunities for younger generation activists, civil society organizations, governments, especially Gender and Foreign Affairs Ministries, the private sector, parliamentarians, celebrities, the UN system, UN Women offices around the world and others to participate in the process.

“Picture it”, an invitation to imagine a world where gender equality has become reality will allow people around the world do engage in activities online but also offline. Moreover, the campaign will be mobilized to contribute in UN Women ‘s areas of comparative advantage through Interagency Network on Women Gender Equality (IANWGE). The UN’s Department of Public Information will be prime partner in pushing out messages. Major online network such as Sheryl Sandberg’s Lean In and Sheryl WuDunn’s and Nicolas Kristof’s Half the Sky movement will be engaged to reach out to their audiences. Private sector involvement will be sought for financial support and to amplify the campaign’s reach (Un women 2014).

3.4 Initial idea - Close the gap

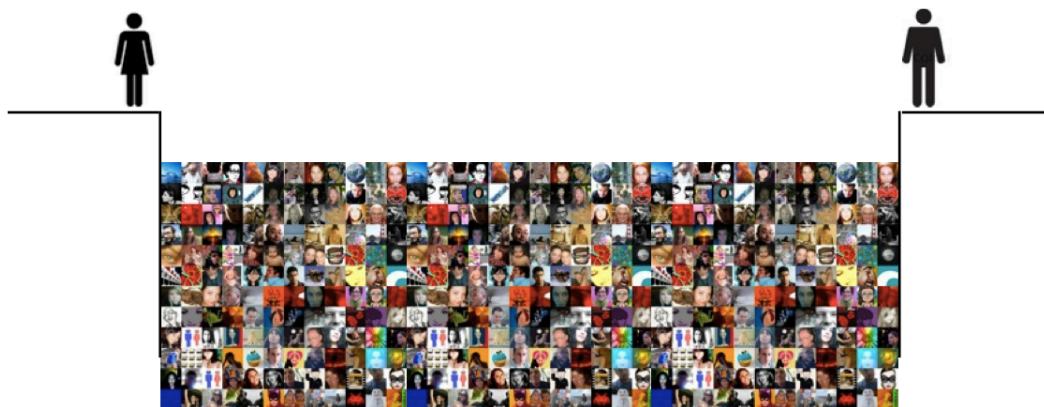


Figure 2: Close the gap - initial idea

Twenty years ago, World leaders met in China to adopt the Beijing Platform for action and make a promise (Beijing Promise). Imagining a bold plan for a world, in which women and men are equal human beings. Shaping their societies and lives side by side. Not two step behind or left out but together. Every women and every girl should have the opportunity to reach her highest dreams. The Beijing promise was not just a dream, it was and it is an

action to reach those opportunities. After twenty years the promise that was made in Beijing have not been fulfilled. Today, UN Women are still playing catch up in the gender equality game. A journey toward women's empowerment and empowering humanity has begun to close the gap in gender equality and keep Beijing Promise alive.

The idea of close the gap results in online visual photo gallery where people fill the gap with their profile pictures one by one, helping women to catch up to men on the other side of the gap (Figure 2).

“Close the gap” is required to enable users to donate their profile pictures for narrowing the gap between women and men in the world. Users can both use their Facebook account and Twitter account or one of them to contribute to the gap. Once users are logged into the social media account, they will be redirected to a thank you message that shows the number of order users be in, and a sharing link to allow users to share the application on their Facebook or Twitter wall.

If users already donated to the campaign, users will be promptly led to a thank you message where users can also share the it on their wall. All users information including names, profile pictures, time, and location will be stored in a database that can be used for fetching and displaying users data on the gap.

Secondly, there should be a language switching bar in the header that let users choose their prefer language such as English, French and Spanish. In which language users are able to share the application on their own language without performing any switching more. In the footer, there are the counter showing the number of users had donated and how much percentage of users over the target number of 20,000 users.

Thirdly, The middle content displays users profile images between the gap of two columns that present for men and women. “1995” in the center of the gap has a meaning of remembering 1995 Beijing Declaration and Platform for Action was adopted. The logos are changed depending on the language users have chosen. By clicking on the logos, users are driven back to the Beijing 20+ main website in according to language.

Finally, since the number of users images can be a huge number, the images should be optimized as much as possible in order to ensure the loading speed of the application. Besides, the other images and text also should be reduced to necessary sized for maximizing the application performance.

Referring to customer requirements document, designing layout and development process should use it as a map to perform efforts. A plan for solutions to above issues should be made before driving to implementation phase. This phase is important in getting agreement of customers. It will save time and effort during the implementation process if developers and customers are on the right track of solutions. The following sub topics will explain the solutions to the application in more detail.

4.1 Facebook Login SDK

Facebook provides a very powerful API to allow the integration of Facebook Login system to the web application. Facebook authentication and sharing are parts of this integration, Facebook SDK for JavaScript offers functionality that can be performed on client side to leverage such integration (Rana 2011). There are certain benefits to using JavaScript SDK since it reduces load bundles on server, and gives better response time as the request goes straight from client browser to Facebook, rather than from browser to server and then Facebook and back again. After users are logged into Facebook account through Facebook authentication system that Facebook JavaScript SDK offers, at this point the SDK will retrieve the applicable Facebook user profile data from Facebook that can be saved to database for later use.

4.2 Twitter API

Similar to Facebook JavaScript SDK, Twitter also provides an API for developers to implement Twitter authentication in their application. “Login with Twitter” is a pattern of authentication that allows users to connect with Twitter using third-party services by just one click.

When users click on “Login with Twitter” button, users are redirected to Twitter. Twitter verifies user credentials, if success users will be redirected to the application with OAuth Token. Using OAuth Token, the application can get the access token from Twitter. Using that access token, the application is able to get users information by making an API call to Twitter (R., K. 2013).

4.3 MySQL

As Carlos states “A database is an integrated computer structure that stores a collection of end-user data and metadata”. Metadata is explained as data about data, it presents data characteristics and data relationships in the database.

#	Column	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(11)			No	None	AUTO_INCREMENT	Change Drop More
2	is_ready	tinyint(1)			No	None		Change Drop More
3	type	tinyint(1)			No	None		Change Drop More
4	some_id	varchar(50)	utf8_swedish_ci		No	None		Change Drop More
5	image	varchar(255)	utf8_swedish_ci		No	None		Change Drop More
6	folder	varchar(255)	utf8_swedish_ci		No	None		Change Drop More
7	name	varchar(255)	utf8_swedish_ci		No	None		Change Drop More
8	gender	varchar(1)	utf8_swedish_ci		No	None		Change Drop More
9	locale	varchar(10)	utf8_swedish_ci		No	None		Change Drop More
10	location	varchar(255)	utf8_swedish_ci		No	None		Change Drop More
11	original_image_url	varchar(255)	utf8_swedish_ci		No	None		Change Drop More
12	created	timestamp			No	CURRENT_TIMESTAMP		Change Drop More

↑ Check All / Uncheck All With selected: Browse Change Drop Primary Unique Index

Figure 3: Database structure of “Close the gap” application

Take the “Close the gap” application as an example, if data includes the user IDs, names, and profile images, metadata stores types of value on each data such as numeric on user IDs, text on names and images (figure 3). Moreover, metadata also provides a whole picture of data in the database that describe the value and use of data.

Data stored in the database are controlled by Database Management System (DBMS) called MySQL. MySQL serves as middleman that interacts between users and the database. Figure 4 depicts that DBMS received application requests from end-users, translates them into a language that database could understand to perform, and then get back the results to end-users.

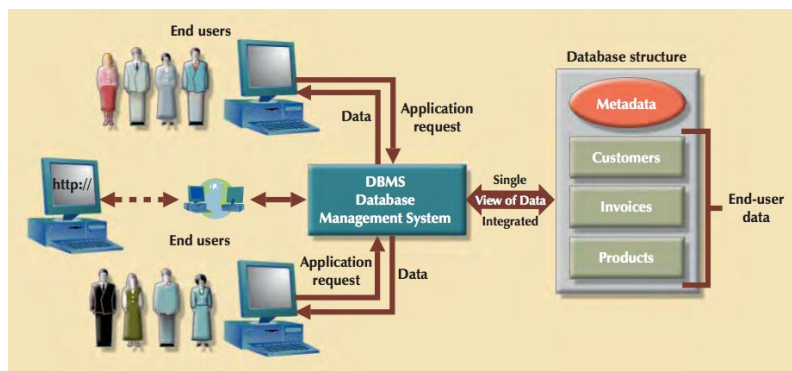


Figure 4: Interaction between users and database via DBMS.

MySQL is one of the most popular DBMS used today, MySQL supports both single user and multiuser. For single user, MySQL allows only one user to interact with database at a time, if other users need to use the database, they must wait for previous user to complete action. For multiuser, MySQL enables multiple of users to perform actions at the same time. In addition, MySQL is classified as a operational database which is designed mainly to support daily operations of a company.

4.4 Languages switching

A languages switching bar to allow users to choose among three languages including English, French and Spanish is implemented using PHP language. First is creating language files that contain the content of each language that are header title, header content, sharing languages, footer content, thanks messages and main content. For example, file [lang.en.php](#) represents the content in English, file [lang.fr.php](#) represents the content in French and file [lang.es.php](#) represents the content in Spanish. However, in order to let browsers understand in which language the content is, language variables must be passed as an HTTP query string to URLs, we are able to get the value of that language variable and switch the language of content accordingly. PHP also supports getting the value of HTTP query string using `$_GET` method. Values of `$_GET` method are superglobals, which mean that they are accessible regardless of the scope (Gabriel 2009). We can access them in any function, class or file anywhere. For more detail, we will show some example of the work in the implementation part below.

4.5 jQuery

jQuery is a JavaScript library which is designed to simplify document navigation, DOM accessing, animations and event handling. jQuery makes the application load smoother by using loading function supported in jQuery library. In addition, some elements of the application need to be scalable on window resizing for better display in small devices, that can only be done using jQuery since the elements can be accessible easily through DOM mechanism (figure 5).

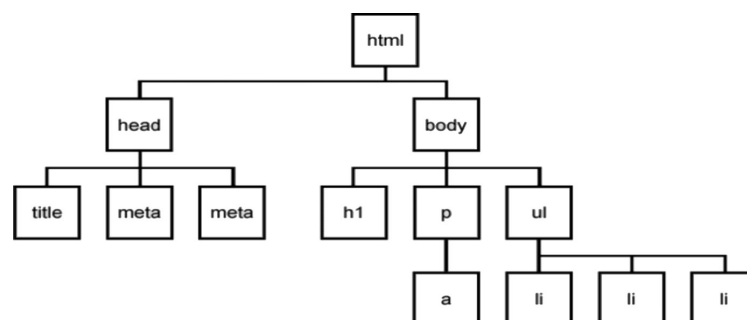


Figure 5: DOM structure

4.6 Auto page loader

jQuery autopager is a plugin that allows to load subpages in a single page automatically on user scrolls downward. This plugin reduces the loading time of the application since the visible content is only loaded as needed. Since the application collects every user contributed, and display all of them in one single page, it must load a huge number of user

profile images. If all images are queried once, it may take infinite time to load the application fully. For that reason, the content should be divided into multiple subpages that are loaded only when user scroll downward.

5 Implementation

5.1 Frontend

Frontend development or another name client-side development is responsible for transferring the design into the interface that user can use. On the other hand, frontend is the interface and backend is the database, these two combines together to make a single application. Each has its own functionality in the application and cannot exist without the other.

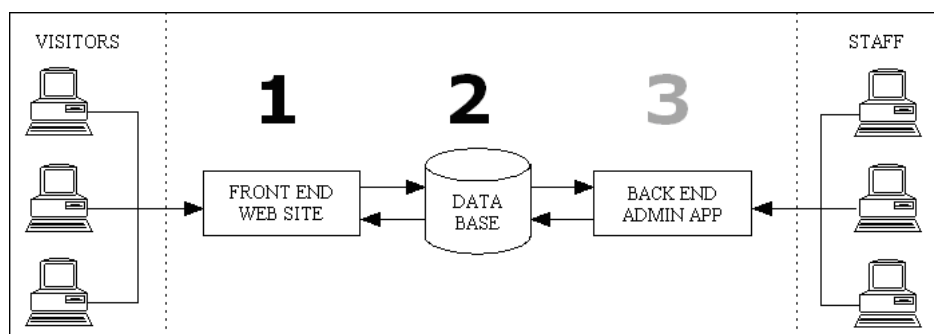


Figure 6: Database connection

Frontend focuses on programming of layout to power the visuals and interactions of the web. If comparing the application as a house, the frontend is the house decoration that makes visitors feel comfortable when visiting the house. The frontend includes the design and the frontend development. The design is performed using Photoshop CS6 and the development is implemented using HTML5 and CSS3, even with jQuery. Frontend makes the combination of HTML5, CSS3 and jQuery become easier for users to control such as fonts, menus, buttons, transitions, sliders, etc (Michael 2013).

Another important issue needs to be considered during frontend development phase is the compatibility of the application in different browsers. Nowadays, Chrome and Firefox are the most popular browsers, however still a group of users use old browser such as Internet Explorer 9 and older which have very low compatibility to common frontend techniques. For that reason, an advice to comfort the old browsers is to create fallback version of development to ensure the application can work properly on the old browsers.

5.1.1 HTML5 / CSS3

5.1.1.1 HTML5

HTML5, a fifth version of HTML standard, is the core technology of Internet markup language responsible for representing content of the World Wide Web (www). Added more features to HTML/XHTML, HTML5 provides not only rich media support, but also creating web applications to interact with users, data, and servers effectively. Thank to this new technology, the code becomes clearer and easier to access than before since HTML5 allows semantic elements to group code, for instance tag <header>, <sidebar>, <section>, <footer>, etc. HTML5 is ready for developing mobile-friendly applications using viewport and home screen icons.

In “close the gap” application, HTML5 is defined by <!DOCTYPE html> with the aim of all browsers could interpret parts of HTML in an HTML5-compliant way. HTML5 should come with declaring character set with <meta charset> to display text in different languages. For example the common character set used is utf8: *<meta charset="utf8">*.

Appendix A, figure 1 shows the HTML5 structures used in the app, including DOCTYPE declaration, tag <HEAD> to define header for the app, and tag <BODY> to define the content of the app.

```
<meta name='viewport' content='width=device-width, initial-scale=1'>
```

Line 9 (See appendix A, figure 1) is supported by HTML5 to make the application compatible to small devices (tablets, smart phones, etc). It ensures that the layout will be always displayed in 1:1 scale. No zooming is applied when the app is running on those small devices.

Structure of body (See appendix A, figure 3) includes a loader to display replacing loading images during the waiting time, so that user will not see the ugly version of the app when all of components are not fully loaded. User can see instead a loading gif image that let they know they have to wait for the app to be loaded.

Secondly, a call-to-action box in the middle of the page to enhance user click on one of the two login buttons “Login with Facebook” or “Login with Twitter”. The content of the box changes depending on how user interact with the app. For instance, if user are not logged into their social media account, the box will show “Fill the gap” to let users know that they should click on the box in order to contribute to the application. If users already contributed, the content of the box will be “You have already shared.” and if users are the first time contributing, the box says “Thank you for sharing! You are the number # to share this application”.

Thirdly, a wrapper displays all the text and content of the application. Wrapper includes a logo of the app, language box to display three language options which are English, Spanish and French, a header that show application title and description, a column presenting icons and images, and a footer showing parameters of total number of registered users, progress bar and sharing buttons.

Appendix A, figure 3 shows the body structure of the application:

5.1.1.2 CSS3

CSS stands for Cascading Style Sheets that define how to represent HTML elements. CSS is stored in files with .css extension, CSS allows appearance and layout of the application to be changed in css files. The application uses three main css files which are style.css determining the whole elements styles through the application, normalize.css responsible for reset styles to their default state and ie.css specially targeting to render the application on Internet Explorer. Appendix A, figure 4 is the structure of CSS3 in “Close the gap” application.

A breakthrough in Internet technology that comes with HTML5 and CSS3 is media queries. Media queries provide support for targeting styles for different browsers on different devices. This brings huge benefit for creating mobile-friendly applications and makes easier job for developers.

The application uses media rules inside style sheet with “screen” media type and “and” operator to target to specific browser based on device width.

Media queries target to some common devices including:

Smartphones (portrait)

```
@media screen and (max-width: 400px)
```

Smartphones (landscape)

```
@media screen and (max-width: 500px)
```

iPad (portrait and landscape)

```
@media screen and (max-width: 800px)
```

Desktop (medium screen)

```
@media screen and (max-width: 1200px)
```

Desktop (large screen)

```
@media screen and (max-width: 1500px)
```

Note: The media queries only define max-width instead of defining both max-width and min-width because the style of smaller max-width can inherit style of its parent. This advantage saves code to be executed and avoids overlapping code.

5.1.2 jQuery

The app uses jQuery as a JavaScript library to position elements properly on both desktop and mobile screens. The elements such as columns, profile images using jQuery to get the right sizes on specific browsers. Browser size is first defined, then other elements are scaled based on the pre-defined size to prevent them from stretching or narrowing.

Declaring variable `wi` for browser width, this variable is global and can be reused.

```
var wi = $(window).width();
```

Normally, the content of webpage will be displayed from left to right, top to bottom. However the app requires the content of user profile images to be rendered reversely. A piece of code (See appendix B, figure 1) is responsible for displaying them from bottom to top, from right to left.

Resize columns according to portion in layout. For example the column-top-left in the layout has portion of 190/1024, that is approximately 0.1855. When displaying on the webpage, the column-top-left should have the width of 0.1855 of window width because when the window is scaled, the width will be changes, then the column-top-left still keeps the same portion at 0.1855 to the window width regardless of scaling. The same formula is applied for column-middle-left and column-middle-right (See appendix B, figure 2).

User profile images have the size of 25x25px each, if the browser width is 1000px, then one row can host 40 images, each image occupies 2.5%. When browser width is only 400px, the each row can only host 16 images, then each image is now 6.25% of browser width. In order to fit the images to browser width, percentage of each image occupies should be calculated based on the image number that each row can host (See appendix B, figure 3).

The following piece of code renders the modal box in the middle of the page by calculating the middle point of browser, then set the position including left and right of modal box to be at the middle point of browser.

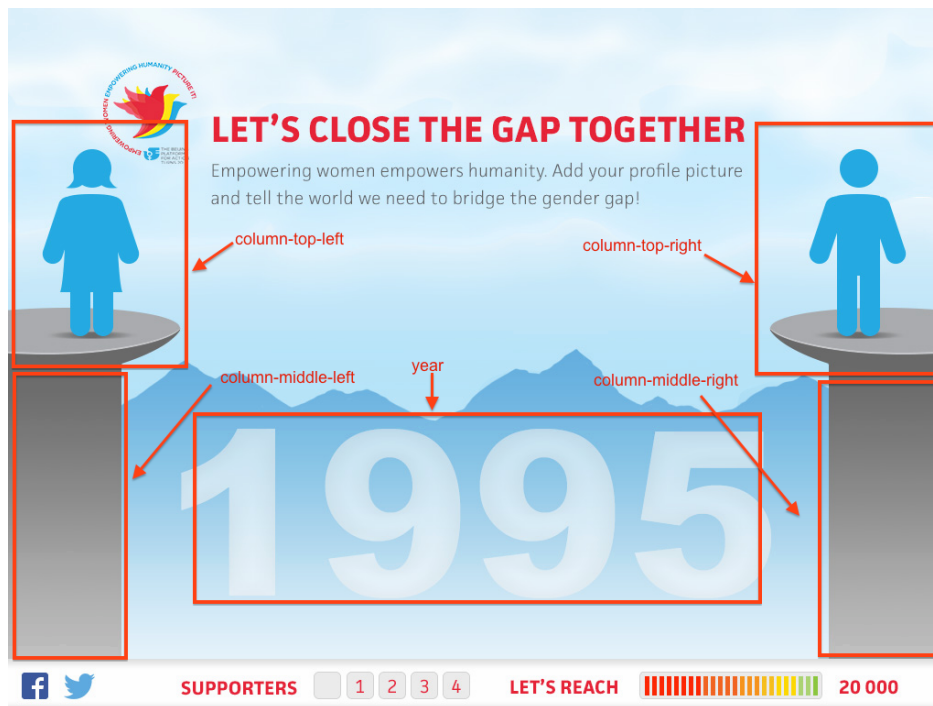


Figure 7: Screenshot of “Close the gap” application

5.2 Backend

As stated above, while frontend is considered as user interface layer, backend works as the database layer to store the data. Frontend development creates the visual presentation using the mixing of HTML, CSS, JavaScript running on web browser. Backend development works with the database to process the data required for creating frontend to users. Backend development deals with security over user authentication, login authentication, and data retrieving. Requests made by users are gone through a security check before being processed, data is looked up in the database and the application is sent to users using a web server (Tony 2013).

Appendix D, figure 2 is a function that uses Facebook API to fetch user detail through an app ID, then returns “thank you” message, otherwise it returns “error” message.

Similarly, the snippet of code (See appendix D, figure 3) shows a function is responsible for getting user detail from Twitter by using Twitter Oath, then returns “thank you message” if function is successful, otherwise it return “error” message.

After retrieving user detail, data is recorded to database using `saveFBData()` and `saveTwitterData()` functions. Three variables including name, gender, locale are defined to store result, then the function executes INSERT INTO statement, and new records will be added to “users” table.

In order to count how many users have already contributed, all users stored in “users” table must be selected. The number of users will be updated accordingly the userIDs counted from the database and stored in the function *updateUserCount*.

```

177 private function updateUserCount()
178 {
179     $cn = $this->getDB();
180     $res = $cn->query('SELECT id FROM user WHERE is_ready=1');
181     $count = $res->num_rows;
182
183     $cn->query('UPDATE config SET data="' . $count . '" WHERE name="user_count");
184     return $count;
185 }

```

To display user profile images on the application, a database request must be made to select all users who have registered. Then using function *getImages()* to push them into an array for later calling.

```

23 public function getImages()
24 {
25     $cn = $this->getDB();
26
27     $res = $cn->query('SELECT * FROM user WHERE is_ready=1 AND image<>"" ORDER BY id DESC');
28     $dataArray = array();
29     $i=1;
30     while($row=$res->fetch_assoc())
31     {
32         $dataArray[$i]='/uimages/' . $row['folder'] . '/' . $row['image'];
33         $i++;
34     }
35
36     return $dataArray;
37 }

```

New users can see thank you message and their order after passing authentication process of Facebook and Twitter, however old users who has contributed can see the message that says “You have already registered” and sharing dialog. In order to execute that function into the application, the backend should has a function *isUserInDb* that selects all users from database, compares the current userID with those in the database, then returns true if userID already exists, returns false if userID does not exist.

```

187 private function isUserInDb($user_id,$type)
188 {
189     $cn = $this->getDB();
190     if($stmt=$cn->prepare('SELECT id FROM user WHERE type=? AND some_id=?'))
191     {
192         $stmt->bind_param('is', $type, $user_id);
193         $stmt->execute();
194         $stmt->store_result();
195         if($stmt->num_rows>0)
196         {
197             $stmt->close();
198             return true;
199         }
200         $stmt->close();
201     }
202     return false;
203 }

```

5.2.1 Database Design

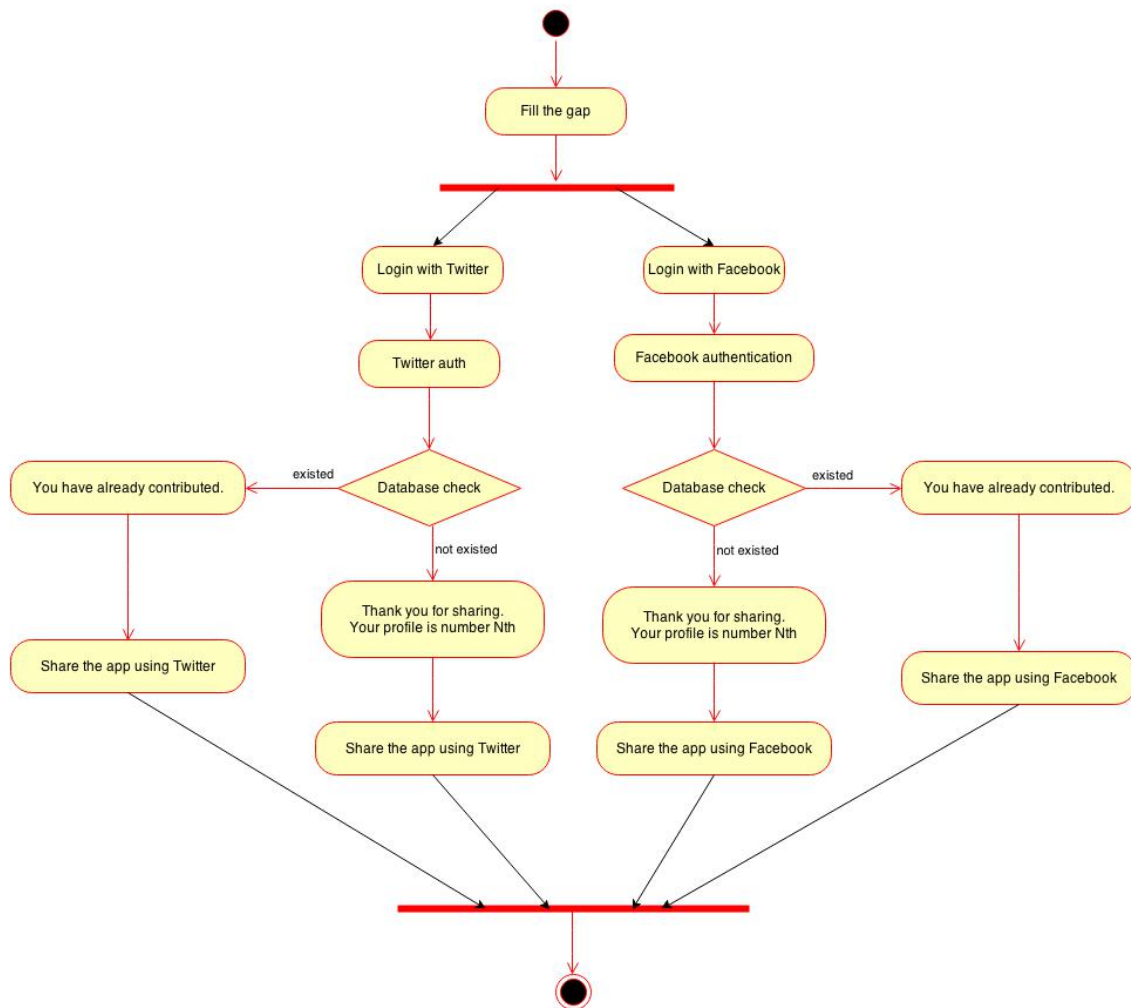


Figure 8: Activity Diagram

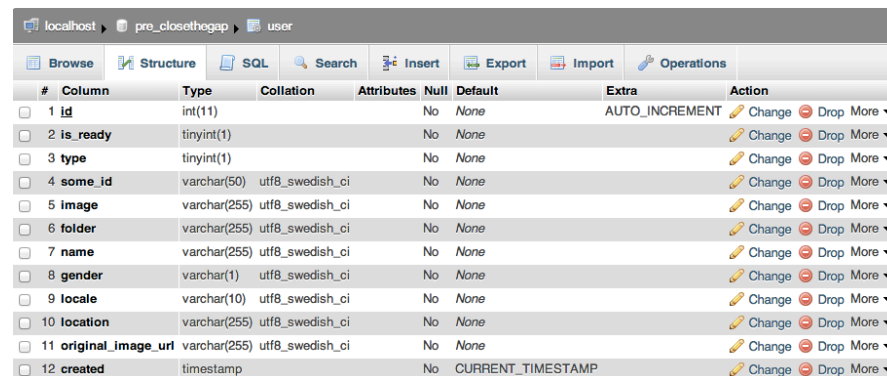
Figure 8 shows that when user go to the application, they can choose one of the following methods which are Fill the gap with Facebook or Fill the gap with Twitter. If the method is using Facebook, user will be redirected to Facebook login dialog. After fill in the information (username and password), Facebook executes authentication to user. If login is successful, user will be checked if they are already in the database. If not, user can see thank you message and are encouraged to share the application with Facebook. If user are already registered, they can see the message that says “You have already registered” and they are also encouraged to share the application with Facebook. If method is using Twitter, the process is similar to Facebook login. The authentication process is performed using Twitter Oauth.

Table and characteristics

In database, a table is considered as a data structure composed of rows and columns to organize information. In this application, the database contains a table user (figure 9) to

store user details including userID, username, profile picture, gender, locale as a series of columns. Each piece of data (such as username) is a field in the table. A set of information (about a particular user) is called records.

In the database, keys are important because they are used to uniquely identify each row in table. Primary key in this application is userID which is unique to each user. Primary key is generated by DBMS and can not be null.



The screenshot shows a database management interface with a table named 'user'. The table structure is as follows:

#	Column	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(11)			No	None	AUTO_INCREMENT	Change Drop More
2	is_ready	tinyint(1)			No	None		Change Drop More
3	type	tinyint(1)			No	None		Change Drop More
4	some_id	varchar(50)	utf8_swedish_ci		No	None		Change Drop More
5	image	varchar(255)	utf8_swedish_ci		No	None		Change Drop More
6	folder	varchar(255)	utf8_swedish_ci		No	None		Change Drop More
7	name	varchar(255)	utf8_swedish_ci		No	None		Change Drop More
8	gender	varchar(1)	utf8_swedish_ci		No	None		Change Drop More
9	locale	varchar(10)	utf8_swedish_ci		No	None		Change Drop More
10	location	varchar(255)	utf8_swedish_ci		No	None		Change Drop More
11	original_image_url	varchar(255)	utf8_swedish_ci		No	None		Change Drop More
12	created	timestamp			No	CURRENT_TIMESTAMP		Change Drop More

Figure 9: Database structure

5.2.2 Database connection

In order to connect to the database, the application must open a connection to MySQL server. In this case, using a supporting function by PHP called `mysqli()` to connect to the MySQL database (figure 20, appendix). The function requires three parameters which are hosting name, MySQL username, the password to login with, and the database name to execute statements.

5.2.3 Facebook SDK

Using JavaScript in web application is extremely common nowadays. Facebook provides an implementation of multiple platforms to access their data. Among the other platforms, JavaScript is very easy to use, integrate and can be used externally with minimal speed effects. Facebook made several functionalities ready to use without requiring much knowledge about coding. The application “Close the gap” implements Facebook authentication using JavaScript SDK. To begin, first registering a Facebook Application named “Close the gap”. Then import the Facebook JavaScript SDK library to the application by the following snippet of code (See appendix C, figure 8).

`FB.init` (See appendix C, figure 8, line 50) is a method triggered on document loading. It initializes the Facebook JavaScript API with custom settings like `appId`, `status`, `cookie` and

xfbml. The “status” parameter tells the Facebook API whether to check the login status or not. The “cookie” parameter allows Facebook API to use cookie for saving session state or not. The session status can be retrieved via cookie variables. This is important since the application is implemented via server code.

The line of code (See appendix C, figure 8, line 68) is responsible for importing Facebook JavaScript library into the application. After initializing the Facebook JavaScript API, a function named `Fb.getLoginStatus()` is called in order to determine whether user is logged into Facebook or not. If the user is already logged, the authentication process is skipped and a thank you message is showed immediately to the user. If not, the user is let through the authentication process like normal.

Check FB.login

`Fb.getLoginStatus()` (See appendix D, figure 1) has function to check if user has been login into Facebook. That function includes a callback inside to return the status of user. If the status is “connected”, user has been passed through the authentication process. Otherwise, a login dialog will be shown up to validate user before user is connected to Facebook.

Feed dialog

The sharing dialog is only visible to user when they pass Facebook authentication process. To utilize the sharing functionality that Facebook JavaScript SDK provides, the same application ID can be used. In order to make the sharing actually work, the application must be public to all users, otherwise the sharing only works for developer’s computer. The sharing dialog is implemented using `FB.ui()` method and jQuery (Figure 23). The function `shareToFB()` has a parameter lang to define which language user prefer using. The sharing dialog can be customized by changing the values of name (the text appears on the dialog), link (the URL to be shared), picture (the url of image to be displayed on the dialog), caption (text displayed under name/link), description (text display under caption), messages (field displayed as a message box).



Figure 10: Facebook Feed Dialog

5.2.4 Twitter SDK

Similar to Facebook, Twitter provides Twitter Login API (See appendix D, figure 3) to embed Twitter login into web application using Twitter OAuth PHP library for making OAuth requests. When user click login button by Twitter, user will be redirect to Twitter. Then Twitter verifies user credentials, if the authentication is successful, user will be redirected to the application using OAuth token by which access token is retrieved. Once access token is get, the application can fetch user data by making an API call to Twitter.

Firstly, consumer key and consumer secret are required from Twitter application to perform the authentication. Secondly, the application must obtain a request token from twitter by sending a signed message to POST/oauth/request_token. Then user is redirected to GET oauth/authentication. If user is already logged in and approved, they will be authenticated and returned to the callback URL within a valid request token. If user is logged in but not approved, they will see a request to share access. Once user accept the request, they will be redirected to the callback URL. If user is not logged in, they will be asked for username and password. After logging in, user will be granted access to the application and returned to the callback URL with valid OAuth request token.

After retrieving valid request token, the application makes a request to POST oauth/access_token. If the request token is passed the oauth_token, a successful response including access token is returned. The token and token secret will be stored and used for future authentication requests to Twitter API.

To validate user, the application uses GET account/verify_credentials. If user has been passed the validation process, they will see a “thank you” message, otherwise they will see an “error” message.

5.2.5 Fetching data

The previous sections described specifically how to access user data from Twitter and Facebook through their provided APIs, and establish a connection to database in order to store the data. In this section, we will discuss about the methodology to query data from database and display them on the user interface.

Firstly, when user click on login buttons, the system will check that whether user already exist in the database, if not then user profiles will be saved into the database, otherwise user will see a thank you message (See appendix C, figure 2).

To count the total number of users have been contributing, another query (See appendix C, figure 7) is performed to get all users from the database, the count the number of them, otherwise return 0 if there is no user in the database.

In order to display the images on the user interface, the system needs to query all the data images from the database and push them into an array. The function `getImages()` (figure 11) returns the array that contains all user profile pictures that exist in the database.

```
public function getImages()
{
    $cn = $this->getDB();

    $res = $cn->query('SELECT * FROM user WHERE is_ready=1 AND image<>"" ORDER BY id DESC');
    $dataArray = array();
    $i=1;
    while($row=$res->fetch_assoc())
    {
        $dataArray[$i]='/uimages/' . $row['folder'] . '/' . $row['image'];
        $i++;
    }

    return $dataArray;
}
```

Figure 11: function to display all user images on user interface

5.2.6 Languages

In order to create a multi-language application, three files containing the text for three languages are created. Three languages text in English, Spanish and French are stored in the files called `lang.en.php`, `lang.fr.php` and `lang.es.php`. Figure 12 is the content of `lang.en.php` file as an example.

```
1 <?php
2 /*
3 Language = English
4 */
5
6 $lang = array();
7
8 $lang['hi'] = 'Let\'s close the gap together!';
9 $lang['p'] = 'Empowering women empowers humanity. Add your profile picture and tell the world we need to bridge the
10 gender gap!';
11 $lang['fill'] = 'Fill the gap !';
12 $lang['number'] = 'Thank you for contributing, your profile picture is nr!';
13 $lang['spread'] = 'Spread the word about the need for women\'s empowerment!';
14 $lang['shareFB'] = 'Share on Facebook!';
15 $lang['shareTW'] = 'Share on Twitter!';
16 $lang['thanks'] = 'Thank you for participating. Share and show your support for equality!';
17 $lang['description'] = 'I\'ve helped close the gender gap, join in building a world without discrimination.
18 Empowering Women. Empowering humanity';
19 $lang['support'] = 'Supporters!';
20 $lang['reach'] = 'Let\'s reach!';
21 $lang['logo'] = 'images/Beijing20_English.png';
22 $lang['return'] = 'Return to the Beijing+20 website';
23 ?>
```

Figure 12: `lang.en.php`

The main file will include a piece of code that gets the right language. When user click to choose which language they prefer, the value of it will be stored in variable `$lang`. Then function `switch()` is responsible for comparing the chosen value with other values and executes according piece of code that belongs to the chosen language. The variable `$lang` is a global variable that can be access through the application code. Moreover, a cookie is used to store the chosen language in thirty days. That means when user come back the application before 30 days, they can still be redirected to the language that they previously selected.

6 Optimizing web performance

Web performance is measured as the page loading time that is the time from a moment that user request a new page to the moment the page is fully rendered by a browser. Optimizing web performance is a method that uses tools and techniques to increase the page loading time. From business point of view, optimizing web performance means to provide users a better experience on visiting the site. Take an online sale web application as an example, the average loading time of an online shopping nowadays is about two seconds, what if customers spend four or five seconds waiting for the site loaded, they will not want to visit that site again. Poor web performance is one of the main reasons make user abandon the website and switch to competitor (Martin 2014).

In order to improve web performance, there are several suggestions that can be implemented during the development process. In this document, both front-end and back-end optimization will be discussed in order to clarify the topic.

6.1 Front-end optimization

The application contains multiple CSS files, each of them is responsible for styling different aspect in the application. For example typography.css takes over the font styling, main.css is used for defining the main styles such as body, content, footer and sidebars, reset.css is important in resetting the styling to its default state, responsive.css makes the application flexible in different browsers, especially for tablets and small devices. By combining those CSS files together, the number of requests that browser has to make will be reduced, so that the page loading time will be decreased accordingly (Kevin 2013).

Minifying CSS and JavaScript files is another suggestion that should be applied to the application. By removing all whitespaces, line breaks, indentation and semicolons at the end of each command, this method helps to minimize overall file sizes and reduce application loading time.

As JavaScript files are much heavier than CSS files, it is recommended that JavaScript files should be loaded after CSS files to avoid delaying browser execution and the download of other assets. The best practice is putting CSS files in the header, and JavaScript files in the footer to ensure that JavaScript files be always loaded finally.

Images are another important element to be considered during the optimization process. By reducing the image sizes and the number of images to be loaded, the application loading time can be reduced effectively. CSS3 has introduced a new technique called CSS Sprites to combine multiple images into one. This technique minimizes the number of images

requested and allows faster asset downloads. Moreover, scaling images to the needed dimensions to avoid browser from resizing the image dimensions on the fly that lengthens the loading time. Images also can be compressed using any tool like Photoshop, Paint to minimize the image sizes.

6.2 Back-end optimization

By using browser caching, the page loading time is reduced by setting expiration time in HTTP headers for static content. The page is no longer needed to load resources via network since it can load directly from local disk. Besides, repeat visitors can be redirected to previous access site by making redirects cacheable. Moreover, connection is one of the most important aspects regarded in the web performance optimization. Choosing fast servers to host the application and the database can help to reduce loading time on user request (Quinn 2014).

In the application, the number of visitors can reach to 20,000. This number is large enough to make the application load 40 times slower than other application if all users in the database are fetched once. A solution to this issue is limiting the number of users to be retrieved from the database to 30 users per query. Then divide the application into multiple subpages. For example, if there are 3000 users contribute to the application, there will be 100 subpages, each subpage contains 30 users.

```
27 $res = $cn->query('SELECT * FROM user WHERE is_ready=1 AND image<>"" ORDER BY id DESC LIMIT 30');
```

By using JQuery autopager plugin, subpages are loaded automatically as the user scroll downwards rather than being loaded at once. As shown in figure 13, `autopager()` has three parameters, `autoLoad` param enables the subpages to be loaded automatically, `content` param determines the content to be loaded using autopager, and `load` param calls a function to trigger window resize when then next subpage was loaded.

```
3 $.autopager({
4   // a selector that matches a element of next page link
5   autoLoad: true,
6   // a selector that matches page contents
7   content: '.profile-image',
8   load: function(current, next) {
9     $(window).trigger('resize');
10  }
11 });})
```

Figure 13: Example of Autopager plugin

6.3 Google Analytics

Google Analytics is added to the application with the purpose of tracking the application's traffic as well as managing the number of people navigating around the application. A simple API (figure 14) provided by Google is free and easy to implement.

```
<script type="text/javascript">
var _gaq = _gaq || []; _gaq.push(['_setAccount', 'UA-3482969-4']); _gaq.push([
'_setDomainName', '.unwomen.org']); _gaq.push(['_trackPageview']); (function() { var ga
= document.createElement('script'); ga.type = 'text/javascript'; ga.async = true; ga.src
= ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') + '.google-
analytics.com/ga.js'; var s = document.getElementsByTagName('script')[0]; s.parentNode.
insertBefore(ga, s); })();
</script>
```

Figure 14: Universal analytics tracking code

7 Final result



Figure 15: Final outcome of Close the gap

After twenty years since the Beijing Conference took place, UN Women has kept Beijing Promise as a target to reduce the gap between women and men, ensure women's rights and empowerment. Close the gap application is introduced in order to raise the awareness of the importance of Beijing Conference Agenda and to gather people in all over the world participating in assuring gender equality. With the support of web technology as well as social media (Facebook, Twitter), close the gap application creates a platform for people in the world contributing their small actions helping reducing the gap of gender and keep the Beijing Promise alive.

By contributing to the application, people help to engage in the campaign close the gap, their pictures will appear between the gap showing that each person donates a small contribution to fill the gap between male and women, erase the distant of gender inequality and protect the right for women. This is also the target that the campaign aims to put Beijing Promise in action.

However, there are still some challenges that the campaign in general and the application in specific might face. Not all the audiences could understand or interest in the meaning that the campaign expected to convey. Besides, some others might think about the rewards or benefits they supposed to be receive after taking part in the campaign. Whether UN Women's messages are successfully transmitted to their audiences is still a question that takes a time for the answer.

8 Project Work and Learning Reflection

The thesis is based on real project work of a media agency and through this paper, working experiences have been shared during the project. Because the project was implemented during the training phase, different know-hows were gained in both user experience field and technical part.

In the beginning period when the project was initialized, several challenges were faced when drafting the first design of the project and getting feedbacks from users. The mockups kept changing many times before generating a final version. During that time, the front-end codes were also updated continuously to fit the user's requirement. Thank for that, knowledge about UX design as well as supporting skills necessary to communicate with customers were gained.

Another difficulty that came during the implementation of the project is a tight schedule. There was only one week to finalizing the demo to the user, and a haft week to be launched. With such a limited time, both front-end and back-end implementation must be managed. Thank to the help from the team, technical skills were improved significantly. Moreover, the needed knowledge about customizing code performance as well as SEO technique were also achieved. In a real work project, coding to make the application works is not enough, the developer should have skill to improve code quality and build a best-performance application.

The knowledge gained from Web Development courses that Laurea offered has a chance to be applied into the real work projects. Although during the working implementation, there were unintentional challenges and bugs occurred, those were managed with the effort and

the support from the team. The project could be considered as a successful result because it received several positive feedbacks from the customer and the user.

9 Conclusions

Began from Beijing Promise made in the World Conference twenty years ago, UN Women - an organization works for ensuring gender equality, determined to launch a campaign that gathers people all over the world contributing their efforts in order to close the gap between women and men. Under the title "Empowering women, empowering humanity, picture it", the campaign aims to regain women' rights and gender equality by sending an invitation to all people to participate in. The initial objective of the campaign results in Close the gap application that allows every people to take part in by donating their social media profile pictures and mobilizes the campaign in a modern technological trend nowadays. The application uses different updating web technology such as HTML5, CSS3, PHP and newest version of jQuery to implement the front-end as well as back-end development. In addition to that, Facebook and Twitter support very important roles in accessing user data thanks to their provided APIs which make the development become easier and more flexible. Besides, techniques that help to optimize the application performance comprise of auto loading data on user scroll down, code validator, code minifying, Google analytics tracking code, etc. All of the process steps from initial idea, planning, designing to technical implementation, testing and maintenance follow a very good progress to make the success of the application. The application now has almost 8600 contributors from all over the world, that number will continuous increasing and the purpose of assuring gender equality will be accomplished in the future.

References

Gabriel, C. 2009. How to add multi-language support to a PHP website. Accessed 28 July 2014. <http://www.bitrepository.com/php-how-to-add-multi-language-support-to-a-website.html>

Google. 2013. Web performance best practices. Accessed 29 July 2014. https://developers.google.com/speed/docs/best-practices/rules_intro

Kevin, M. 2013. The ultimate checklist for optimizing web performance. Accessed 29 July 2014. <http://www.yottaa.com/blog/bid/296075/Infographic-The-Ultimate->

Checklist-for-Optimizing-Web-Performance. Martin, T. How to write a thesis proposal. Accessed 6 July 2014. http://www.ldeo.columbia.edu/~martins/sen_res/how_to_thesis_proposal.html

Rana, M. 2011. Introduction to Facebook JavaScript SDK. Access 28 July 2014. <http://codesamplez.com/development/access-facebook-graph-api-using-javascript-sdk>

Quinn, M. 2014. How to optimize your website's performance and keep customers happy. Accessed 28 July 2014. <http://www.forbes.com/sites/theyec/2014/02/19/how-to-optimize-your-websites-performance-and-keep-customers-happy/>

Michael, T. 2013. Front-end vs Back-end Development. Accessed 29 July 2014. <http://itknowledgeexchange.techtarget.com/itanswers/front-end-vs-back-end-development/>

Ravishanker, K. 2013. How Twitter OAuth works. Access 28 July 2014. <http://hayageek.com/login-with-twitter/>

Tony, M. 2013. Why you should build your web application back-to-front. Accessed 26 July 2014. <http://www.tonymarston.net/php-mysql/back-to-front.html>

DDB Helsinki Copywriter. 2014. UN Women First. Access 1 October 2014.

UN Women Copywriter. 2014. Campaign_UN Women_2014. Accessed 1 October 2014.

UN Women Copywriter. 2014. N.K.Beijing_Promise_L. Accessed 1 October 2014.

UN Women Copywriter. 2014. UNWomen_Beijing+20 campaign concept. Accessed 1 October 2014.

Web design process. 2010. The University of Texas at Austin. Accessed 6 July 2014. <https://www.utexas.edu/learn/designprocess/>

List of figures

Figure 1: Software Development Life Cycle	8
Figure 2: Close the gap - initial idea	12
Figure 3: Database structure of “Close the gap” application.....	15
Figure 4: Interaction between users and database via DBMS.	15
Figure 5: DOM structure	16
Figure 6: Database connection	17
Figure 7: Screenshot of “Close the gap” application	21
Figure 8: Activity Diagram	23
Figure 9: Database structure	24
Figure 10: Facebook Feed Dialog.....	25
Figure 11: function to display all user images on user interface.....	27
Figure 12: lang.en.php.....	27
Figure 13: Example of Autopager plugin	29
Figure 14: Universal analytics tracking code	30
Figure 15: Final outcome of Close the gap	30

Appendix A : HTML Structure

Figure 1: Application HTML5 structure

```

1 <!DOCTYPE html>
2 <!--[if lt IE 7]> <html class="no-js lt-ie9 lt-ie8 lt-ie7"> <![endif]-->
3 <!--[if IE 7]> <html class="no-js lt-ie9 lt-ie8"> <![endif]-->
4 <!--[if IE 8]> <html class="no-js lt-ie9"> <![endif]-->
5 <!--[if gt IE 8]><!--> <html class="no-js"> <!--<![endif]-->
6 <head>
7 <meta charset="utf-8">
8 <title>UN Women's</title>
9 <meta name='viewport' content='width=device-width, initial-scale=1'>
10 </head>
11 <body>
144
145 </body>
146 </html>

```

Figure 2: Header screenshot

```

1 <!DOCTYPE html>
2 <!--[if lt IE 7]> <html class="no-js lt-ie9 lt-ie8 lt-ie7"> <![endif]-->
3 <!--[if IE 7]> <html class="no-js lt-ie9 lt-ie8"> <![endif]-->
4 <!--[if IE 8]> <html class="no-js lt-ie9"> <![endif]-->
5 <!--[if gt IE 8]><!--> <html class="no-js"> <!--<![endif]-->
6 <head>
7 <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
8 <title>UN Women's</title>
9 <meta name='viewport' content='width=device-width, initial-scale=1'>
10
11 <link rel="stylesheet" href="css/style.css" />
12 <link rel="stylesheet" href="css/normalize.css" />
13 <link rel="stylesheet" href="css/ie.css" />
14
15 <!-- Load Font -->
16 <link rel="stylesheet" href="http://webfonts.justanotherfoundry.com/Facit/" type="
text/css" />
17
18 <!--script src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8.23/jquery-ui.min.js"><
/script-->
19 <script src="js/jquery.min.js"></script>
20 <script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script>
21
22 <script src="js/plugins.js" type="text/javascript"></script>
23 <script src="js/main.js" type="text/javascript"></script>
24
25 <!--[if IE]><script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script><![
endif]-->
26 </head>

```

Figure 3: Body screenshot

```

27 <body>
28 <div class="loader"></div>
29
30 <div id="boxes">
39 </div> <!--.boxes-->
40
41
42 <!-- Wrapper -->
43 <div class="wrapper">
44
45 <div class="logo">
46 <a href="http://beijing20.unwomen.org/<?php echo $_SESSION['lang'];?/" title="<?php
echo $lang['return'];?>" target="_top"></a>
47 </div>
48
49 <header>
50 <h1><?php echo $lang['h1']; ?></h1>
51 <p><?php echo $lang['p']; ?></p>
52 </header>
53
54 <!-- Language box -->
55 <div id="LangBox">
72 </div>
73
74 <!-- Columns -->
75 <div class="column-top">
78 </div>
79 <div class="column-content">
91 </div><!--.column-content-->
92
93 <!-- Footer -->
94 <footer>
113 </footer>
114
115 </div><!--.wrapper-->
116
117 </body>

```

Figure 4: CSS structure screenshot

```

/*----- Header -----*/
header {
}
h1 {
}
/*----- Modal box -----*/

#mask {
}

#boxes .window {
}

/*----- Language Box -----*/
#langBox {
}
#langBox ul li {
}
#langBox ul li a {
}
#langBox ul li:last-child a {
border-right: 0;
}
#langBox ul li a:hover {
color: #ff0036;
}
#langBox ul li.active a {
color: #ff0036;
}
}

```

Appendix B: Javascript functions

Figure 1: Reverse images function

```

$.fn.reverseChildren = function() {
return this.each(function(){
var $this = $(this);
$this.children().each(function(){ $this.prepend(this) });
});
};
$('.profile-images').reverseChildren();

```

Figure 2: Scaling column images

```

var posColumn = function () {
var wi = $(window).width();
var x = ( wi * 190 ) / 1024;
var x_bottom = ( wi * 125 ) / 1024;
var y = ( x * 235 ) / 190;

$('.column-top-left').width(x);
$('.column-middle-left').width(x_bottom);
$('.column-middle-right').width(x_bottom);
$('.column-middle-center').width(wi - (2*x_bottom));
$('.column-top').height(y);
$('.column-top-left').height(y);
$('.column-top-right').width(x);
$('.column-top-right').height(y);

if ( wi < 1024 ){
$('.year img').width( wi * 0.647646 );
}
}

```

Figure 3: Function fitImage() to scale images fit the gap area

```

var fitImage = function() {
var wi = $(window).width();
var x_bottom = ( wi * 125 ) / 1024;
var imageNumber = Math.ceil(( wi - ( 2*x_bottom ) ) / 25);
var imageWidth = (100 / imageNumber);
$('.profile-image').css('width', imageWidth + '%');
}

```

Appendix C: PHP functions

Figure 1: Function to retrieve user information from Facebook

```

60 public function getDataFromFacebook($user_id,$token,$lang)
61 {
62     $facebook = new Facebook(array(
63         'appId' => FB_APP_ID,
64         'secret' => FB_APP_SECRET,
65     ));
66     $facebook->setAccessToken($token);
67
68     try {
69         $user_profile = $facebook->api('/me');
70     } catch (FacebookApiException $e) {
71     };
72
73     if(isset($user_profile))
74     {
75         if(!$this->isUserInDb($user_id, 1))
76         {
77             $this->saveFBData($user_id,$user_profile);
78             $count = $this->updateUserCount();
79             return array('success'=>true,'count' => $count);
80         }
81         return array('success'=>false,'msg'=>$lang['thanks']);
82     }
83     return array('success'=>false,'msg'=>'There were problems in connecting to the Facebook.
84     Try again, please.');
```

Figure 2: Function to retrieve user information from Twitter

```

39 public function getDataFromTwitter($lang)
40 {
41     if(isset($_SESSION['access_token']))
42     {
43         $access_token = $_SESSION['access_token'];
44         $connection = new TwitterOAuth(TWITTER_API_KEY, TWITTER_API_SECRET, $access_token['
45         outh_token'], $access_token['oauth_token_secret']);
46         $userArr = $connection->get('users/show', array('screen_name' => $access_token['
47         screen_name']));
48         if(isset($userArr) && $userArr->screen_name == $access_token['screen_name'])
49         {
50             if(!$this->isUserInDb($access_token['user_id'], 2))
51             {
52                 $this->saveTwitterData($userArr);
53                 $count = $this->updateUserCount();
54                 return array('success'=>true,'count' => $count);
55             }
56             return array('success'=>false,'msg'=>$lang['thanks']);
57         }
58     }
59     return array('success'=>false,'msg'=>'There were problems in connecting to the Twitter.
60     Try again, please.');
```

Figure 3: Function to save Facebook data to the database

```

138 private function saveFBData($user_id,$user_profile)
139 {
140     $cn = $this->getDB();
141
142     $name = (isset($user_profile['name'])?$user_profile['name']: '');
143     $gender = (isset($user_profile['gender'])?$user_profile['gender']: '');
144     $locale = (isset($user_profile['locale'])?$user_profile['locale']: '');
145     $type=1;
146     $is_ready=1;
147
148     if($stmt=$cn->prepare('INSERT INTO user (is_ready,type,some_id,name,gender,locale) VALUES
149     (?,?,?,?,?)'))
150     {
151         $stmt->bind_param('iissss', $is_ready, $type, $user_id, $name, $gender, $locale);
152         $stmt->execute();
153         $id = $stmt->insert_id;
154         $stmt->close();
155
156         $basefolder = dirname(__FILE__) . '/../uimages/';
157         $filename=strtolower(md5(time() . rand(1,9999))) . $id . '.jpg';
158         $folder = substr($filename,0,2);
159         if(!is_dir($basefolder . $folder))
160         {
161             mkdir($basefolder . $folder);
162         }
163
164         $fb_image_url = 'https://graph.facebook.com/' . $user_id . '/picture';
165
166         $img = file_get_contents($fb_image_url);
167         file_put_contents($basefolder . $folder . '/' . $filename, $img);
168
169         if($stmt=$cn->prepare('UPDATE user SET image=?,folder=?, original_image_url=? WHERE id=
170         ?'))
171         {
172             $stmt->bind_param('sssi', $filename, $folder,$fb_image_url,$id);
173             $stmt->execute();
174             $stmt->close();
175         }
176     }
177 }
```

Figure 4: Function to save Twitter data to the database

```

90 private function saveTwitterData($user_profile)
91 {
92     $user_profile = (array) $user_profile;
93     $cn = $this->getDB();
94
95     $user_id = (isset($user_profile['id'])?$user_profile['id']: '');
96     $name = (isset($user_profile['name'])?$user_profile['name']: '');
97     $gender = (isset($user_profile['gender'])?$user_profile['gender']: '');
98     $locale = (isset($user_profile['lang'])?$user_profile['lang']: '');
99     $location = (isset($user_profile['location'])?$user_profile['location']: '');
100     $type=2;
101     $is_ready=1;
102     if($stmt=$cn->prepare('INSERT INTO user (
103         is_ready,type,some_id,name,gender,locale,location) VALUES (?, ?, ?, ?, ?, ?)'))
104     {
105         $stmt->bind_param('iissss', $is_ready, $type, $user_id, $name, $gender, $locale, $
106             location);
107         $stmt->execute();
108         $id = $stmt->insert_id;
109         $stmt->close();
110
111         $twitter_image_name=$user_profile['profile_image_url'];
112
113         if($twitter_image_name!='')
114         {
115             #
116         }
117     }
118 }
119 }

```

Figure 5: Function `getDb()` to make connection to the database

```

205 private function getDB()
206 {
207     if($this->cn===null)
208     {
209         $this->cn = new mysqli(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);
210         $this->cn->query('SET NAMES utf8');
211     }
212
213     return $this->cn;
214 }
215 }

```

Figure 18: System checking

```

if(isset($user_profile))
{
    if(!$this->isUserInDb($user_id, 1))
    {
        $this->saveFBData($user_id,$user_profile);
        $count = $this->updateUserCount();
        return array('success'=>true,'count' => $count);
    }
    return array('success'=>false,'msg'=>$lang['thanks']);
}
}

```

Figure 6: Function to check if user exist in the database

```

private function isUserInDb($user_id,$type)
{
    $cn = $this->getDB();
    if($stmt=$cn->prepare('SELECT id FROM user WHERE type=? AND some_id=?'))
    {
        $stmt->bind_param('is', $type, $user_id);
        $stmt->execute();
        $stmt->store_result();
        if($stmt->num_rows>0)
        {
            $stmt->close();
            return true;
        }
        $stmt->close();
    }
    return false;
}
}

```

Figure 7: function to count number of user images

```

public function getImageCount()
{
    $cn = $this->getDB();
    $res = $cn->query('SELECT * FROM config WHERE name="user_count"');
    if($res->num_rows==1)
    {
        $row=$res->fetch_assoc();
        return $row['data'];
    }
    return 0;
}

```

Figure 8: languages.php

```

1 <?php
2 session_start();
3 header('Cache-control: private'); // IE 6 FIX
4
5 if(isset($_GET['lang'])) {
6     $lang = $_GET['lang'];
7
8     // register the session and set the cookie
9     $_SESSION['lang'] = $lang;
10
11     setcookie('lang', $lang, time() + (3600 * 24 * 30));
12 } else if(isset($_SESSION['lang'])) {
13     $lang = $_SESSION['lang'];
14 } else if(isset($_COOKIE['lang'])) {
15     $lang = $_COOKIE['lang'];
16 } else {
17     $lang = 'en';
18 }
19
20
21 switch ($lang) {
22     case 'en':
23         $lang_file = 'lang.en.php';
24         break;
25     case 'fr':
26         $lang_file = 'lang.fr.php';
27         break;
28     case 'es':
29         $lang_file = 'lang.es.php';
30         break;
31     default:
32         $lang_file = 'lang.en.php';
33 }
34 }?>
35 <input type="hidden" id="sess_var" value="<?php echo $_SESSION['lang']; ?>"/>
36 <?php include_once 'properties/'.$lang_file;
37 ?>

```

Appendix D: Facebook / Twitter APIs

Figure 1: Facebook API connection

```

142 $('#fb_connect').click(function(){
143     FB.getLoginStatus(function (response) {
144         if (response.status == 'connected') {
145             FBParticipate(response.authResponse.userID, response.authResponse.accessToken);
146         } else {
147             FB.login(function (response) {
148                 if (response.status == 'connected') {
149                     FBParticipate(response.authResponse.userID, response.authResponse.accessToken);
150                 }
151             });
152         }
153     });
154 });

```

Figure 2: Get Facebook login status

```

47 <div id="fb-root"></div>
48 <script type="text/javascript">
49     window.fbAsyncInit = function() {
50         FB.init({
51             appId      : '<?php echo FB_APP_ID?>',
52             status      : true,
53             cookie      : true,
54             xfbml       : true
55         });
56
57         FB.getLoginStatus(function (response) {
58             if (response.status == 'connected') {
59                 FBParticipate(response.authResponse.userID, response.authResponse.accessToken);
60             }
61         });
62     };
63
64     (function(d){
65         var js, id = 'facebook-jssdk';
66         if (d.getElementById(id)) {return;}
67         js = d.createElement('script'); js.id = id; js.async = true;
68         js.src = "//connect.facebook.net/en_US/all.js";
69         d.getElementsByTagName('head')[0].appendChild(js);
70     })(document);
71 </script>

```

Figure 3: Twitter API connection

```

4  if(!isset($_SESSION['access_token']))
5  {
6
7      $connection = new TwitterOAuth(TWITTER_API_KEY, TWITTER_API_SECRET);
8
9      $request_token = $connection->getRequestToken(TWITTER_OAUTH_CALLBACK);
10
11     $_SESSION['oauth_token'] = $token = $request_token['oauth_token'];
12     $_SESSION['oauth_token_secret'] = $request_token['oauth_token_secret'];
13
14     /* If last connection failed don't display authorization link. */
15     switch ($connection->http_code) {
16         case 200:
17             /* Build authorize URL and redirect user to Twitter. */
18             $url = $connection->getAuthorizeURL($token);
19             header('Location: ' . $url);
20             break;
21         default:
22             /* Show notification if something went wrong. */
23             echo 'Could not connect to Twitter. Refresh the page or try again later.';
24     }
25 }
26 else{?>

```

Figure 4: Get data from Twitter

```

39  public function getDataFromTwitter($lang)
40  {
41      if(isset($_SESSION['access_token']))
42      {
43          $access_token = $_SESSION['access_token'];
44          $connection = new TwitterOAuth(TWITTER_API_KEY, TWITTER_API_SECRET, $access_token['oauth_token'], $access_token
45          ['oauth_token_secret']);
46          $userArr = $connection->get('users/show', array('screen_name' => $access_token['screen_name']));
47          if(isset($userArr) && $userArr->screen_name == $access_token['screen_name'])
48          {
49              if(!$this->isUserInDb($access_token['user_id'], 2))
50              {
51                  $this->saveTwitterData($userArr);
52                  $count = $this->updateUserCount();
53                  return array('success'=>true,'count' => $count);
54              }
55              return array('success'=>false,'msg'=>$lang['thanks']);
56          }
57          return array('success'=>false,'msg'=>'There were problems in connecting to the Twitter. Try again, please.');
```

Figure 5: Validating user and displaying messages

```

174  function TwitterParticipate(type)
175  {
176      if(type==1)
177      {
178          $.ajax({
179              type: "post",
180              url: "/participate_twitter.php",
181              dataType: "json",
182              success: function(json){
183                  if(json.success)
184                  {
185                      $('#dialog').html(
186                          '<div class="thank box"><div class="close_box">X</div><p><span class="contribute">' + number +
187                          '</span>' + json.count + '<span class="spread">' + spread +
188                          '</span></p><p id="share_to_twitter"> ' + share +
189                          '</p></div>');
190                      $('#share_to_twitter').click(function() {
191                          shareToTwitter(lang);
192                      });
193                  }
194                  else
195                  {
196                      $('#dialog').html(
197                          '<div class="joined box"><div class="close_box">X</div><p>' + json.msg +
198                          '</p><p id="share_to_twitter"> ' + share +
199                          '</p><p class="return"><a href="http://beijing20.unwomen.org/' + lang + '" target="_top">' + link +
200                          '</a></p></div>');
201                      $('#share_to_twitter').click(function() {
202                          shareToTwitter(lang);
203                      });
204                  }
205              }
206          });
207      }
208      else
209      {
210          //twitter connect failed
211      }
212  }
213  }

```