



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Lam Nguyen Chanh

ENHANCING STEPOUT MOBILE APPLICATION WITH REACT NATIVE

Technology and Communication
2020

ACKNOWLEDGEMENT

First of all, I would like to express his appreciation towards all of Vaasa University of Applied Sciences authorities (VAMK), and especially Dr. Ghodrat Moghadampour – the author’s supervisor of this thesis project, for their kind support and enthusiasm during his study time for the bachelor’s degree.

Moreover, the author is also thankful for all the teamwork which has been carried out with the StepOut management department and development team, whose guidance has encouraged the author to elevate his work over technical difficulties.

Finally, the author wishes to display emotional gratefulness to his friends and family, those who have stood by his side from the beginning of this journey, being his mental support and the inspiration to continue on this journey overall and his career path in particular.

Helsinki, January 8, 2021

Lam Nguyen Chanh

ABSTRACT

Author	Lam Nguyen Chanh
Title	Improving and Developing new features of StepOut Mobile App with React Native
Year	2020
Language	English
Pages	64
Name of Supervisor	Ghodrat Moghadampour

The thesis was carried out with the purpose of maintaining an existing StepOut mobile application codebase, which was originally developed using React Native technology. The Stepout mobile application is a well-being application which aims to help its users to maintain good health and improve quality of their daily lives. It provides browsing and booking functionalities for the fitness activities of their desire and motivate them to workout frequently. Furthermore, the aim was to develop brand new features for City of Helsinki innovation project.

All the new features were delivered within this thesis have made this version considered the StepOut mobile application first Minimum Viable Product (MVP). Those were a subscription package model, a point and rewards system and push notification feature on Android platform.

The result of this thesis is a new version of the application which has met the initial requirements, and an overall positive review (80% approximately) from the group of users from the City of Helsinki, who have been experimented with StepOut mobile application and service from the beginning of the innovation project.

Keywords	StepOut mobile app, React Native, front-end development, Javascript, cross-platform mobile development, fitness
----------	---

CONTENTS

ACKNOWLEDGEMENT	2
LIST OF FIGURES AND TABLES	5
LIST OF CODE SNIPPETS.....	7
LIST OF ABBREVIATIONS.....	8
1 INTRODUCTION	9
1.1 Project Overview.....	10
1.2 Objectives.....	10
2 RELEVANT TECHNOLOGIES	13
2.1 React Native versus Native	14
2.2 Summary	17
3 PROJECT DESCRIPTION	19
3.1 Requirement Specifications	19
3.2 Working Environment Configurations and Implementations	22
3.2.1 General Communication Channels and Documentations.....	22
3.2.2 Codebase Version Control.....	22
4 IMPLEMENTATION.....	25
4.1 Codebase Inheritance and GitHub workflow establishment.....	25
4.2 StepOut Access – Subscription Model Implementation	28
4.3 Push Notification on Android.....	41
4.4 Implementation of Reward System	43
4.5 Overview	51
5 DEPLOYMENT PROCEDURES.....	53
5.1 iOS	53
5.2 Android.....	57
6 CONCLUSIONS	61
REFERENCES	63

LIST OF FIGURES AND TABLES

Figure 1. Visual comparison between React Native and Native Development. /8	18
Figure 2. Explore screen display in the StepOut mobile application is available for both non-logged in and logged in users.....	20
Figure 3. Visualization of workflow with forks in GitHub.	24
Figure 4. An example of a GitHub issue with the “feature request” type.....	27
Figure 5. A pull request created related to the issue.	28
Figure 6. StepOut Access introducing screen design on XD.	29
Figure 7. StepOut Access introducing screen establishment on StepOut application.....	30
Figure 8. Payment methods slide-up sheet.	31
Figure 9. Payment methods screen.	32
Figure 10. Payment methods handling modal.....	32
Figure 11. English localization in-app.	34
Figure 12. Finnish localization in-app.	34
Figure 13. StepOut Access OrderConfirmation screen at first glance.	35
Figure 14. Slide up modal box and input field to enter access code.	36
Figure 15. Informative modal box UI when StepOut Access package successfully purchased.....	37
Figure 16. Asynchronous and Synchronous programming. /11/.....	38
Figure 17. Life cycle of an application with Redux. /12/	39
Figure 18. Class Detail when user have StepOut Access.	40
Figure 19. Purchasing class when having StepOut Access.....	41
Figure 20. Firebase console UI and push notification creating procedure.....	42
Figure 21. Push notification on Android.	42
Figure 22. The new appearance of StepOut Access intro screen.	43
Figure 23. Reward screen implementation.	44

Figure 24. Policy Update Notification screen.	46
Figure 25. New policy revision.	47
Figure 26. User privacy consent given.	48
Figure 27. Many City of Helsinki users have updated the application via TestFlight.	49
Figure 28. Out-dated template for booking confirmation email.	50
Figure 29. Updated template for booking confirmation email.	51
Figure 30. General configurations for iOS project.	53
Figure 31. Apple developer certificates and application capability configurations.	54
Figure 32. iOS build action for archiving.	54
Figure 33. Distribution (deployment) options.	55
Figure 34. Apple distribution certificate signed and ready to upload.	55
Figure 35. iOS distribution has been initiated.	56
Figure 36. Build has been uploaded to Apple Connect successfully.	56
Figure 37. iOS new update submission for Apple review.	57
Figure 38. Android version modification.	58
Figure 39. Execute “./gradlew app:assembleRelease” for building a release APK file.	59
Figure 40. Locating generated APK file.	59
Figure 41. An approved production-ready build on Google Play.	60
Table 1. User stories of StepOut application, with a separation between the old version and the current one.	11
Table 2. Requirements of StepOut mobile app first MVP	21

LIST OF CODE SNIPPETS

Code Snippet 1. An example of a poor-quality code portion exists in StepOut mobile application codebase.	26
Code Snippet 2. Localization for English version.....	33
Code Snippet 3. Localization for Finnish version.....	33

LIST OF ABBREVIATIONS

MVP	Minimum Viable Product
UI	User Interface
API	Application Programming Interface
EU GDPR	Europe General Data Protection Regulation

1 INTRODUCTION

According to numerous studies in recent years, the health and fitness marketplace in European countries has grown drastically and constantly despite the impact of the Covid-19 crisis on the general economic situation and has not shown the sign of being slow down. /1/

These particular present circumstances led to the trend of many large international chains of gyms, equipment manufacturing companies to form and “jump” into the promising business field, with different kinds of business models. But these seem to be usually expensive, with limited choices of fitness activities, with strict subscription models, as well as membership bindings. Although, many companies have realized these are potential obstacles for their business model and therefore, they have tried to offer more flexibility to their clients and related parties. However, there are still countless drawbacks in this marketplace that are still possibility for improvements. /2/

LetStepOut is an early age start-up that is based in Helsinki, eager to target this shortcoming of the marketplace to offer an innovative solution. Their objective is to support independent trainers who just had a fresh start, with a suitable skillset, looking for expanding their businesses. LetStepOut aims to provide a platform acting as a connecting “bridge” between these trainers and their potential customers, according to what these customers are demanding. Therefore, StepOut would be able to create a learning and developing environment for both trainers and customers, without establishing any forceful binding between both sides, so that everyone can be happy and healthy, with the desire of stepping out of their comfort zone. /3/

1.1 Project Overview

Taking advantage of the existed drawbacks of Finland's fitness marketplace, the StepOut mobile application was initially created with the intention of giving the people who love training the flexibility of their own training choice. In more detail, to offer them browsing and booking training sessions (both on-location and online) which are offered by trainers who are co-operating with StepOut. Furthermore, with the on-boarding of an in-house software development team, the City of Helsinki's innovation project was an opportunity for LetStepOut Oy to offer an even more flawless service experience to the users, which were a subscription package, getting along with a point system embedded in rewarding system, and a push notification feature for Android platform.

This thesis report is separated into two main parts: The first part provides an overview to the background of React Native and a dedicated comparison between React Native and Native development, in which circumstances React Native should be picked over Native and on the other hand, where would be the best place for Native development. The second part is a project implementation diary of the author when maintaining and developing new features on the StepOut mobile application.

1.2 Objectives

The main objective of this thesis was to improve the already existing StepOut mobile application and to extend it further with the new features.

The user stories of StepOut mobile application are demonstrated in the table below.

Table 1. User stories of StepOut application, with a separation between the old version and the current one.

As a ...	Story	
User	Old version	Current version
	Information of all the sessions can be browsed with and without logging in.	I can register for a StepOut Access subscription cycle by applying a coupon code.
	I can book for single session and make the payment each time. (require login)	With a StepOut Access subscription, I can book for free any 12 sessions within the current cycle.

	I can login into the app with 3 rd party login which are Google and Apple (on iOS).	With StepOut Access, any sessions I have attended, I can earn a point from it. With points, I can claim rewards which are offered from StepOut.
	I can register and login as a normal user with an id and password.	I can cancel the booking 24 hours prior the session scheduled time to have my session allowance refunded
	I can cancel booked sessions and receive the refund	I can receive push notification on StepOut Android application

2 RELEVANT TECHNOLOGIES

For this project, the key technology used was React Native.

React Native is an open-source Javascript-based cross-platform mobile application framework that allows developers to build native-rendered mobile applications on both iOS and Android platforms with only one codebase. React Native was built based on React, which is a well-known Javascript library amongst the Front-end developers' community, so this translates to the fact that Front-end developers who previously worked with web applications will be able to make a transition to Mobile App Development with React Native much easier and then create strong, robust mobile applications in a short amount of time. /4/

React Native was first introduced and released by the Facebook engineering team as an open-source project in 2015, and soon it became one of the most efficient mobile applications development solution for start-up and Entrepreneurs, who are newcomers to the marketplace and do not have strong financial funds to develop two separate native applications on both iOS and Android. Despite React Native has its own pros and cons, which will be mentioned in detail later, various top of the game mobile applications are still being powered by it, such as Instagram, Facebook, Skype and Discord. /4/

React Native was chosen to develop StepOut mobile applications for these reasons:

- Time and financial saving. /5/
- Shortening the development cycle. Javascript is so familiar with a majority of developers, therefore, recruiting new developers will not cause major issues and time consuming to the development process. /5/

- Framework updates and supports will be ensured, get rid of the concern about the technology will be outdated after a couple of years. /5/
- React Native renders native views and components, unlike some other cross-platform frameworks out there, such as Cordova, Capacitor, PhoneGap, etc. render Web views after the codebase finishes compiling. This explains why the performance on React Native is outstanding compare to the others. /6/

2.1 React Native versus Native

React Native seems to be a rising star in the Mobile App Development world, some individuals have even gone as far as saying that it is on the way to take over the pre-eminence of Native Mobile App Development in the near future. /7/

In order to draw a conclusion about this statement, the author has conducted a deep research to determine the pros and cons of both React Native and Native developments.

- **Pros of React Native:**

- Time-efficiency.
- Hot reloading: The results of a developer's work can be monitored in approximately real-time, without having to recompile the code and re-building the app. /7/
- Cost-efficiency: The savings money comes from having only one development team instead of two, this also means easier project management during the development procedure and more control over deadlines. /7/
- Uniformity of the application on both iOS and Android: In reality, iOS and Android development teams will never be able to work at

the same pace and release a new version of the application at the same time. However, with React Native, the progress is most likely always at the same pace for both platforms. /7/

- Attracts more potential clients: With the fact that React Native development will potentially save a significant amount of time and money, more and more clients will look to take advantage of this.

- **Pros of React Native:**

- Lower quality and performance: Unlike native applications, React Native applications cannot fully utilize the capabilities of the operating systems and devices. Therefore, React Native is not a good choice for applications with heavy, complicated animations, and especially not for mobile games. /7/
- Support and technical issue fixes, a stable version has not been released yet. Therefore, new errors in React Native will take time to address and figure out a proper solution from the community. /7/
- Difficult to implement native features into apps: Even though React Native renders native components when building the application, some native features can be extremely difficult to implement or require different configurations between two platforms, for example, push notifications and card payments.
- Longer time for new features: One codebase for both Android and iOS means that sometimes, developers have to specify the logic for one platform and another way around. For example, the development team would like to implement “Login with Apple” functionality on the iOS platform, however, it is impossible to implement this particular task on Android, therefore, one effective solution is to only display the button which triggers that “Login with Apple” function only if the end device is an iOS one. In general, there

will be a various combinations of possibilities similar to the example to take place, then it would require logic separation and longer time to properly test the application on both platforms before launching new updates. /7/

- **Pros of Native Application Development:**

- Parental support and security from platform providers (Apple and Google): Crucial support from these gigantic companies give native development the advantage of being dependable. Similarly, considering the security aspect, native development is capable of utilizing full hardware power to process an appropriate amount of protection layers to the user's private data. /7/
- High performance: As mentioned, native development allows both devices and operating systems being able to utilize full capabilities, which include the highest achievable framerate, speed, computing power, and graphics, to mention a few. /7/

- **Cons of Native Application Development:**

- Time-consuming and high development cost: Compared to React Native, finance presumably is the first and major aspect that constrains early-age businesses to think about alternative solutions. Developing two separate native applications for iOS and Android requires two independent development teams, which means double wage. /7/
- Complication on managing two products: Product owners will be obliged to work with two development teams at the same time, which brings out more concerns and matters twice as much as managing a single product team. /7/

2.2 Summary

Overall React Native is a decent and valuable alternative solution to native development. However, Native Development still leads the race when the priority comes to quality and performance, the advantages of native development are unquestionable.

React Native would be a beneficial technology to develop mobile applications under some circumstances since the profit comes from a lower price and shorter development time enhances its attractiveness.

All in all, React Native matches every aspect that an early age start-up such as LetStepOut Oy needed: low cost, time-efficient development, with a not too complex a business model for having issues with long-term application scalability.

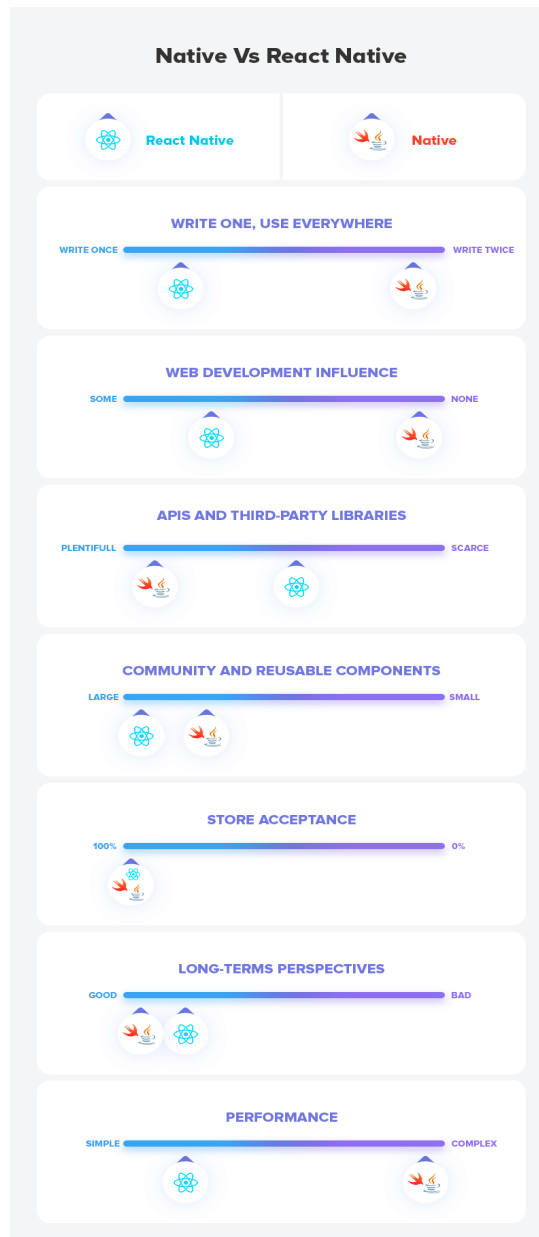


Figure 1. Visual comparison between React Native and Native Development. /8

3 PROJECT DESCRIPTION

In this section, workflow convention as well as the requirements of the new features is demonstrated and explained.

3.1 Requirement Specifications

The core concepts of the StepOut mobile application are granting information access to everything related to StepOut's service, including exploring workout sessions, providing information, details about each particular session, and trainer.

Before the codebase was transferred to StepOut's Development team, including the author, the StepOut mobile application gave the consumers capability to browse the content of the application without logging in, which means for example taking a glance at what is all the training sessions, which one is suitable for them, whether it is an online session or an on-location session, and information about the trainers.

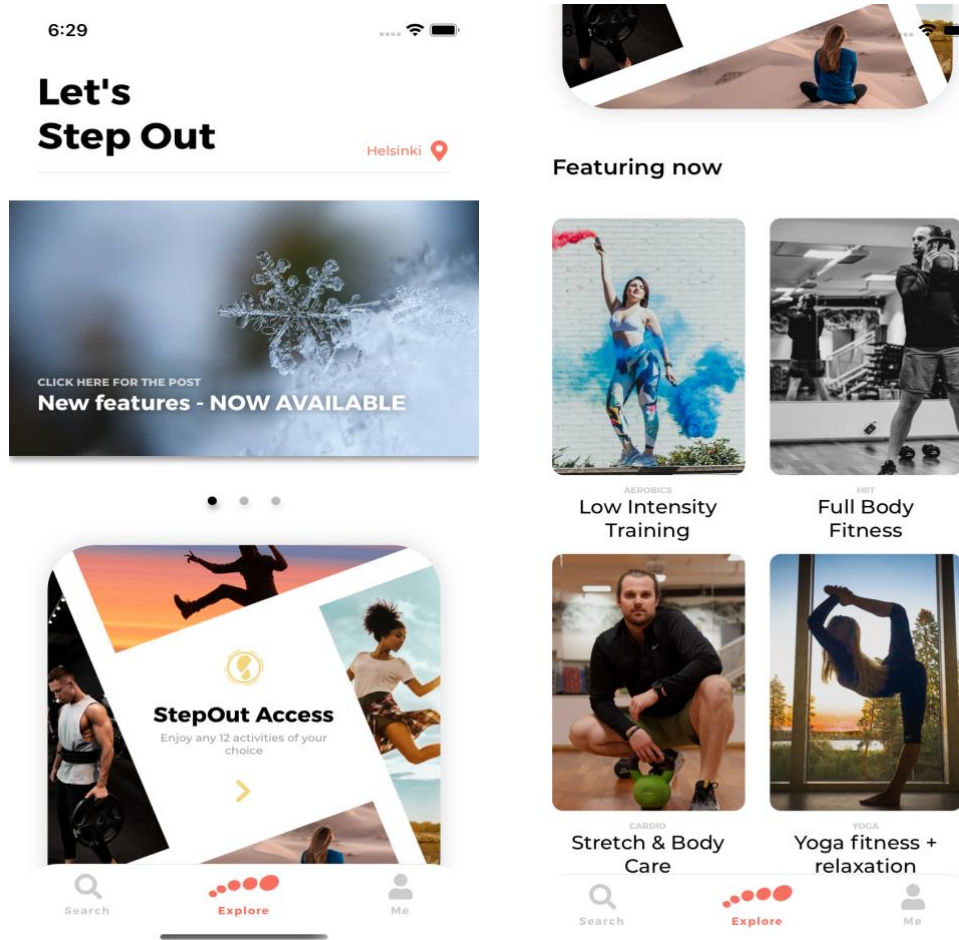


Figure 2. Explore screen display in the StepOut mobile application is available for both non-logged in and logged in users.

At the beginning of September 2020, LetStepOut Oy had the opportunity to collaborate with the City of Helsinki on the Innovation project, with the specified requirements followed by the agreement between both parties, LetStepOut has determined to work on an MVP which will introduce a brand-new subscription model and experiment an innovative rewards system to encourage users on being worked out more often.

After consideration, the StepOut team devised a two phases roadmap, the first phase ended after one month, approximately from 14 September 2020 to 19 October 2020, with the development of subscription model related features, then the second phase was conducted from the end of phase one until 23 November 2020, with the implementation of rewards system and notification.

The table below visualizes the requirements and priority of each one of them.

The priorities respectively are:

1. Must have
2. Should have
3. Nice to have

Table 2. Requirements of StepOut mobile app first MVP

Requirement description	Priority (1 = highest, 3 = lowest)
Displaying information about StepOut Access subscription package.	1
Functionality for users to purchase StepOut Access package.	1
Functionality for users to monitor their StepOut Access package status after purchasing (expiry date, points, etc..)	1
Functionality for users to claim the rewards depends on the number of points they acquired when attending	1

workout sessions with StepOut Access package	
Push notification function for notifying users in several circumstances	2
Responsive support for devices with viewport width narrower than 360 dp	3

3.2 Working Environment Configurations and Implementations

In this section, a daily basis workflow and deployment procedure for mobile applications will be revealed, including the author's experience and personal thoughts.

3.2.1 General Communication Channels and Documentations

On a daily basis working flow, Slack has been used for communication between team members when it comes to work relating matters and Notion for documenting important objects such as meeting notes, roadmap, working progress, and individual tasks.

3.2.2 Codebase Version Control

StepOut is a mobile application with a decent complexity of business logic, therefore, a proper way of understanding what is going on in the code base is crucial. Therefore, GitHub was decided to be StepOut's version control system, due to its outstanding well-establishment, affordability, and rich support.

In the beginning, it was mandatory that a separate repository must be “forked” from StepOut’s GitHub company account into each developer's personal account. Forking means making a copy of the original repository, with an unseen digital link back to that original one. The next step in the procedure is cloning the forked repository to each developer's local machine, and start setting up remote, feature branches, then commit and push new code onto it. /9/

The reason why forked repository is utilized although it seems like making everything more complicated is cutting down the possibility of confusion that might be caused by mutating the original repository directly, for example, producing merge conflicts for other developers on the team. /9/

Later on, when developers would like to submit changes to the original repository, a pull request is beneficial for this particular objective; the pull request should be reviewed by other team members and only apply if it is approved by a team leader or a person who is capable of making decision. Similarly, new changes can be fetched to one’s forked repository by using the command “git fetch upstream” or “git fetch upstream/<branch-name>”.

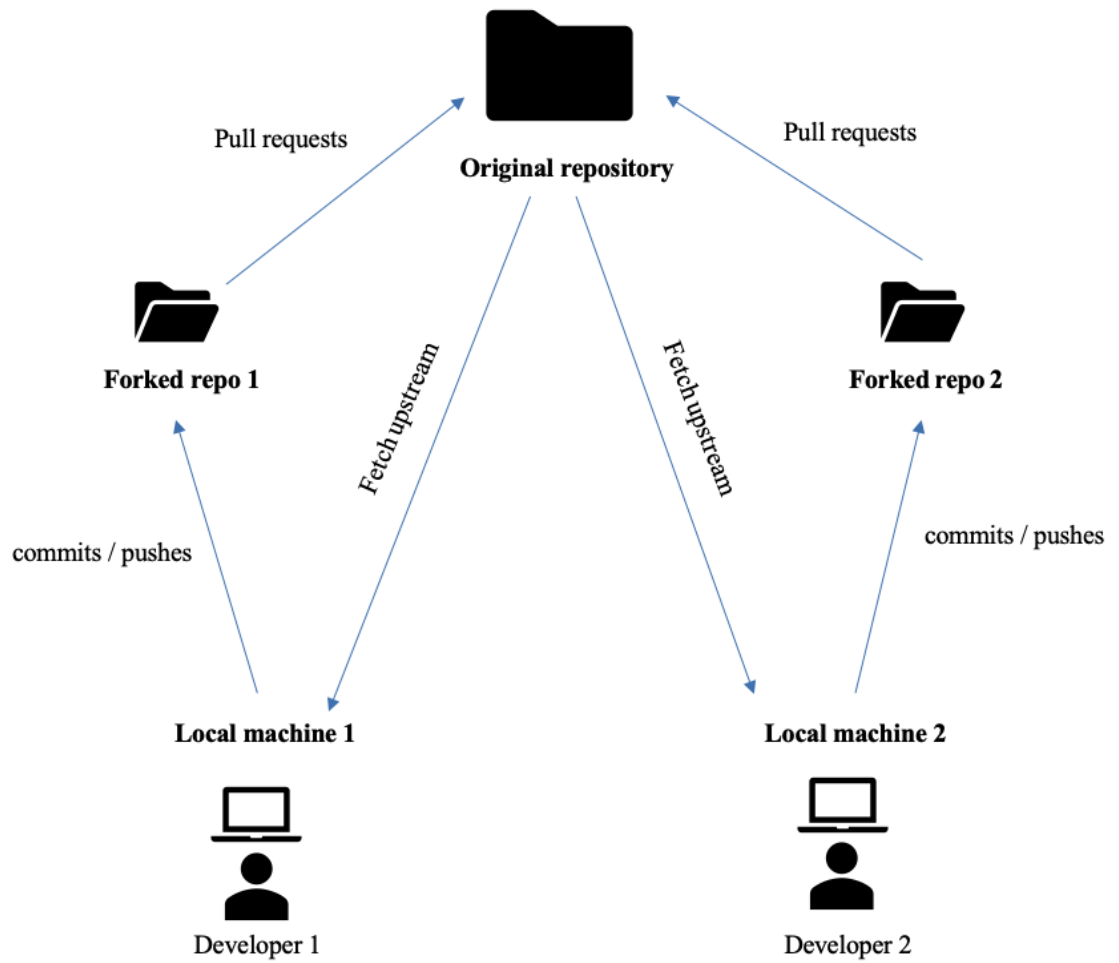


Figure 3. Visualization of workflow with forks in GitHub.

4 IMPLEMENTATION

4.1 Codebase Inheritance and GitHub workflow establishment

In the first working week, the study of the codebase was started by reading React Native official documentations, then putting effort to pin point the code base remaining issues and refactoring those, including left over “console.log()”, redundant comments, poor code formatting, reorganize the project infrastructure.

```

signOut = async () => {
  try {
    const isSignedIn = await GoogleSignin.isSignedIn();
    if (this.state.isNormalLogin) {
      GLOBAL_METHODS.savePref(CONST_PREFERENCES.is_logged_in_
                              _with_normal, false);
      GLOBAL_METHODS.savePref(CONST_PREFERENCES.is_logged_in_
                              with_explore, false);

      this.reset();
    } else if (this.state.isAppleLogin) {
      this.signOutWithApple();
      GLOBAL_METHODS.savePref(CONST_PREFERENCES.is_logged_in_
                              with_apple, false);
    } else if (isSignedIn) {
      // await GoogleSignin.revokeAccess();
      await GoogleSignin.signOut();
      GLOBAL_METHODS.savePref(CONST_PREFERENCES.is_logged_in_
                              with_google, false);

      this.reset();
    } else {
      await LoginManager.logout();
      GLOBAL_METHODS.savePref(CONST_PREFERENCES.is_logged_in_
                              with_fb, false);

      this.reset();
    }
  }
  // var isSignedIn = await GoogleSignin.isSignedIn()

```

```

// GLOBAL_METHODS.getPref(CONST_PREFERENCES.is_logged_in_
                        with_normal).then(
// (this.setState( {isNormalLogin: true } ))
// )
} catch (error) {
  console.error(error);
} signOut = async () => {
  try {
    const isSignedIn = await GoogleSignin.isSignedIn();
    if (this.state.isNormalLogin) {
      GLOBAL_METHODS.savePref(CONST_PREFERENCES.is_logged_in_
                              with_normal, false);
      GLOBAL_METHODS.savePref(CONST_PREFERENCES.is_logged_in_
                              with_explore, false);

      this.reset();
    } else if (this.state.isAppleLogin) {
      this.signOutWithApple();
      GLOBAL_METHODS.savePref(CONST_PREFERENCES.is_logged_in_
                              with_apple, false);
    } else if (isSignedIn) {
      // await GoogleSignin.revokeAccess();
      await GoogleSignin.signOut();
    }

  } catch (error) {
    console.error(error);
  }
}

```

Code Snippet 1. An example of a poor-quality code portion exists in StepOut mobile application codebase.

During this time, a working convention was created with other developers, those respectively were:

- Creating a GitHub issue prior to making any changes to the source code. There will be three types of issues: bug report, custom issue, and feature request. Each of these will have a particular template in order to standardize the procedure.

- On the developer's local machine, check out the current codebase to another branch using the "git checkout -b <number-of-created-issue>" command.
- After the developer's work is done and commits are ready to be merged, pull requests should be created to merge the changes from the developer's forked repository to the project main repository. During the creation of a pull request, a clear description that explains the context and related to which issue must be specified. This action is intended for further issue tracking in the future in case of something wrong takes place.

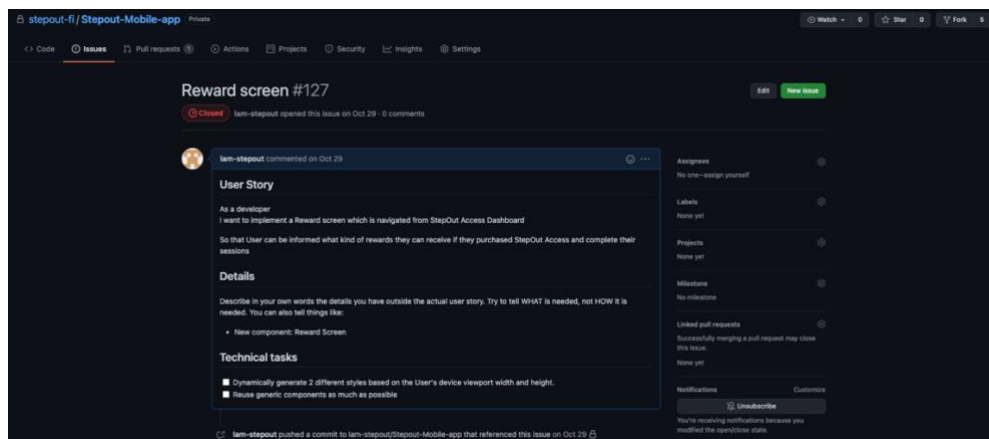


Figure 4. An example of a GitHub issue with the "feature request" type

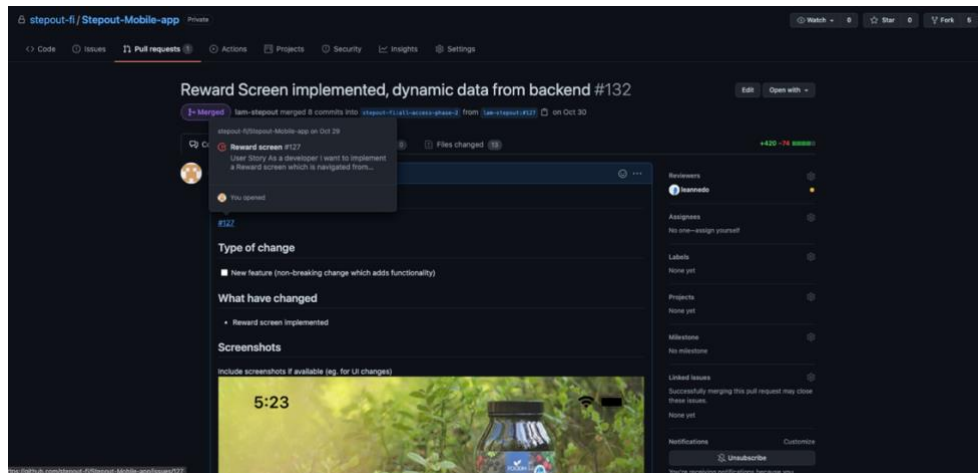


Figure 5. A pull request created related to the issue.

4.2 StepOut Access – Subscription Model Implementation

This feature marked down the first major feature of the innovation project MVP, which is a subscription package granting the privilege for a user to attend any 12 sessions provided by StepOut service free of charge within a cycle of 30 days.

Starting with StepOut Access introducing screen, based on the UI design of StepOut’s product designer on Adobe XD, the design was transformed into the application UI component, thenceforward, navigation into the new component was also established.

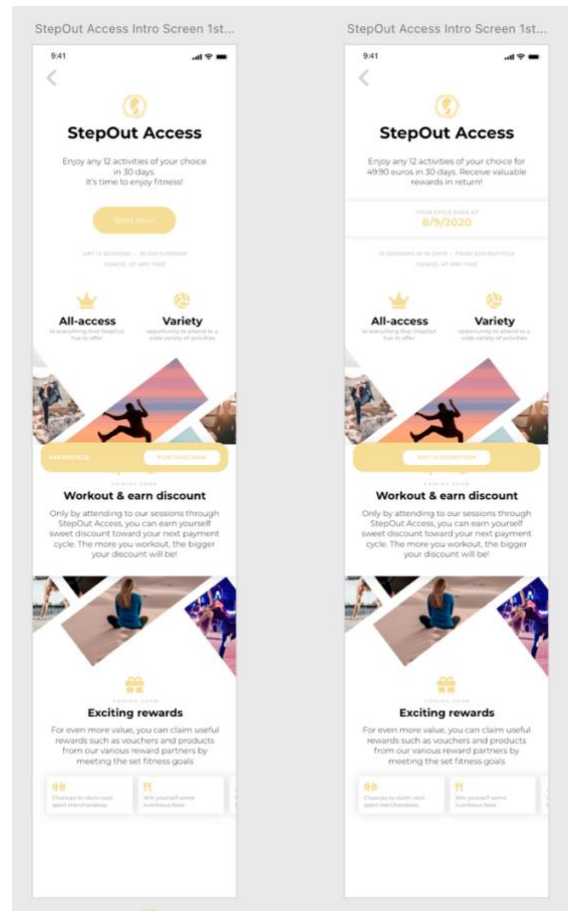


Figure 6. StepOut Access introducing screen design on XD.

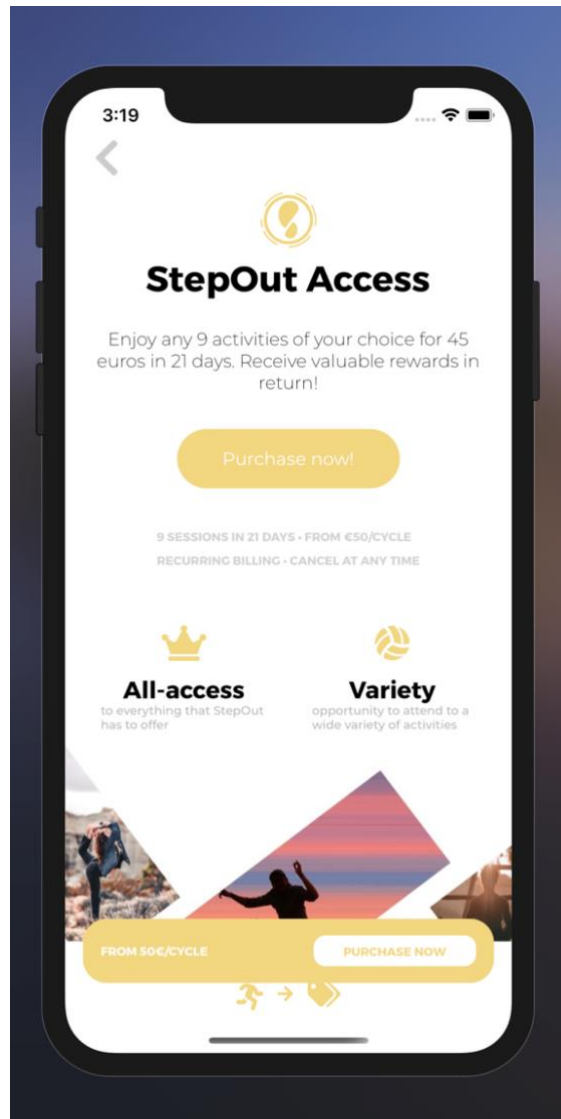


Figure 7. StepOut Access introducing screen establishment on StepOut application.

Following the StepOut Access introducing screen completion was the implementation of UI components related to payment methods management.

This UI implementation included a payment methods screen, slide up sheet, and a pop-up modal. However, these components at this point were UI skeleton only, functionality has not yet to be implemented.

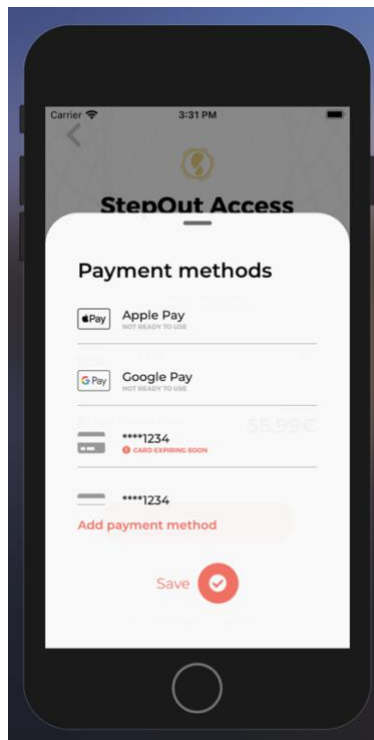


Figure 8. Payment methods slide-up sheet.

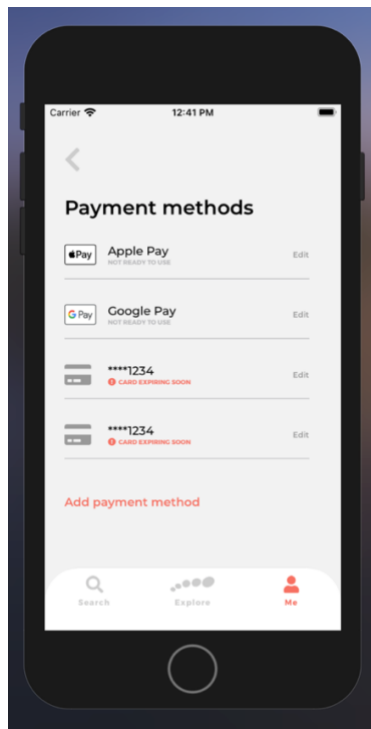


Figure 9. Payment methods screen.

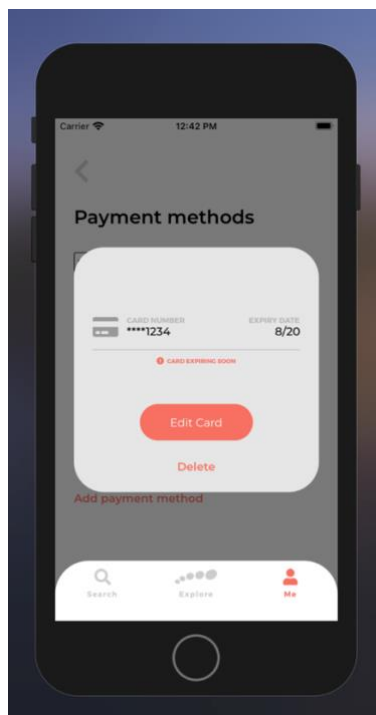


Figure 10. Payment methods handling modal.

The next action points were adding language localizations for recently implemented components, and the most important was the implementation of the StepOut Access package order screen.

Localization is essential for StepOut application since the international customer segmentation and marketplace were targeted in the first place, which in particular, English and Finnish are supported as in the first MVP. The implementation was quite straightforward thanks to the I18n library, and localization will be programmatically specified based on the language of a user device.

```
orderConfirmationScreen: {
  cycleStartFromText: `CYCLE STARTS FROM`,
  subscriptionDescription1: ` for `,
  subscriptionDescription2: ` sessions per `,
  subscriptionDescription3: ` days cycle`,
  cycleEndText: `Cycle ends at: `,
  enterAccessCodeText: `ENTER ACCESS CODE`,
  validAccessCodeText: `VALID ACCESS CODE`,
```

Code Snippet 2. Localization for English version.

```
orderConfirmationScreen: {
  cycleStartFromText: `JAKSO ALKAA`,
  subscriptionDescription1: ``,
  subscriptionDescription2: `:sta tunnista `,
  subscriptionDescription3: ` päivän jaksossa`,
  cycleEndText: `Jakso päättyy: `,
  enterAccessCodeText: `SYÖTÄ PÄÄSYKOODI`,
  validAccessCodeText: `PÄÄSYKOODI`
```

Code Snippet 3. Localization for Finnish version.

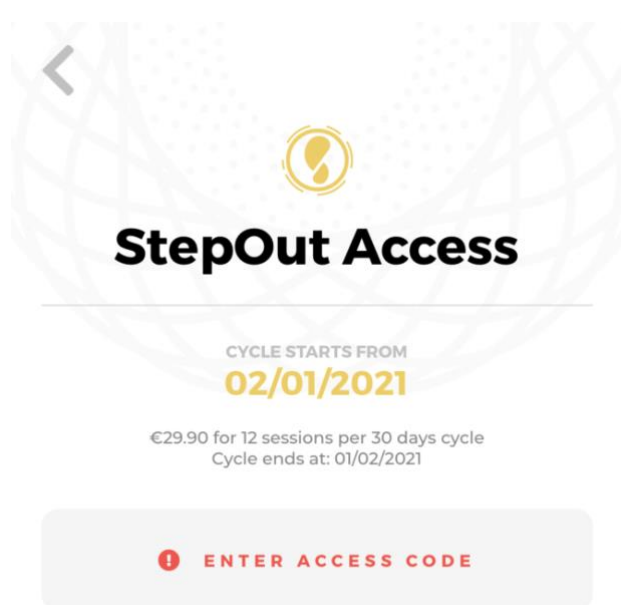


Figure 11. English localization in-app.

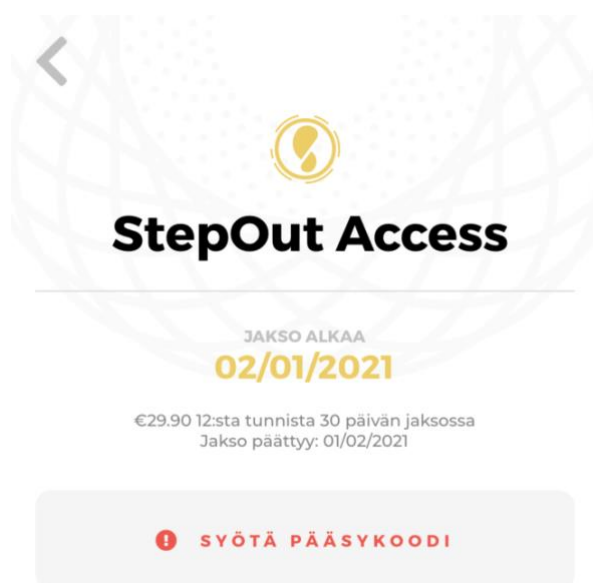


Figure 12. Finnish localization in-app.

The consecutive task was implementing the StepOut Access OrderConfirmation screen. User experience played a massive role for a user to become a long-term user of StepOut service; therefore, it was taken care of with extra attention, every step of action should be obvious and effortless so that any user would be able to subscribe to the package. However, at the early stage of the MVP, StepOut Access could be purchased with a valid coupon code, not with ordinary card payment methods. The reason for this functionality incapacity is the StepOut mobile app using Stripe to handle online payment, however any payment within European Economic Area (EEA) that exceeds the 30€ price threshold would have to be executed with Strong Customer Authentication (SCA) as part of the second Payment Services Directive (PSD2) /10/, and the library that React Native use to handle payment – “tipsi-stripe”, have not been updated to be able to execute precisely meeting the requirement.

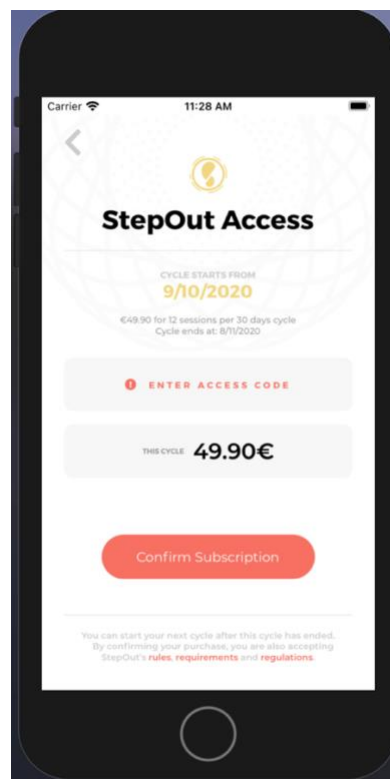


Figure 13. StepOut Access OrderConfirmation screen at first glance.

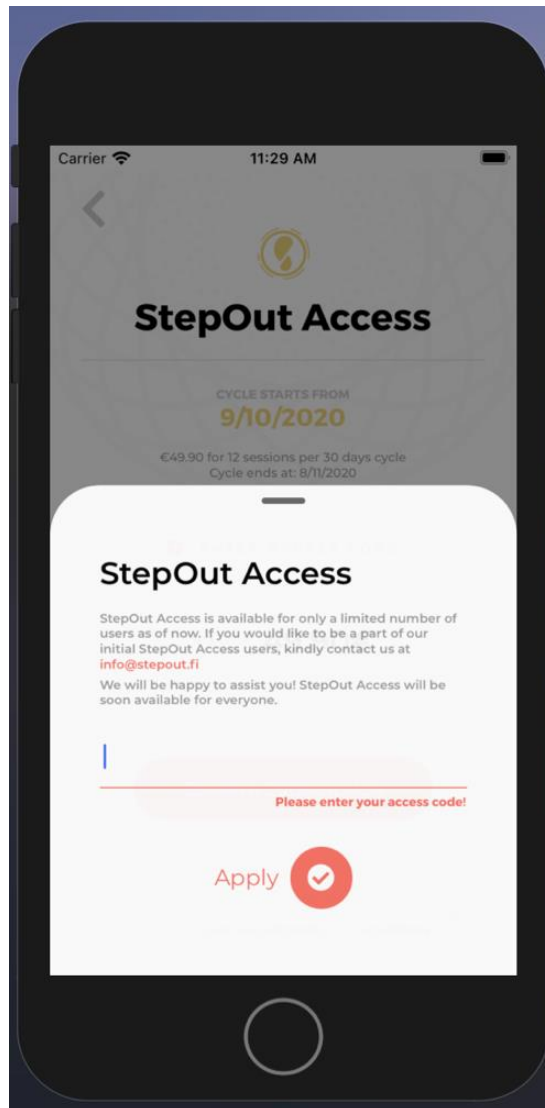


Figure 14. Slide up modal box and input field to enter access code.

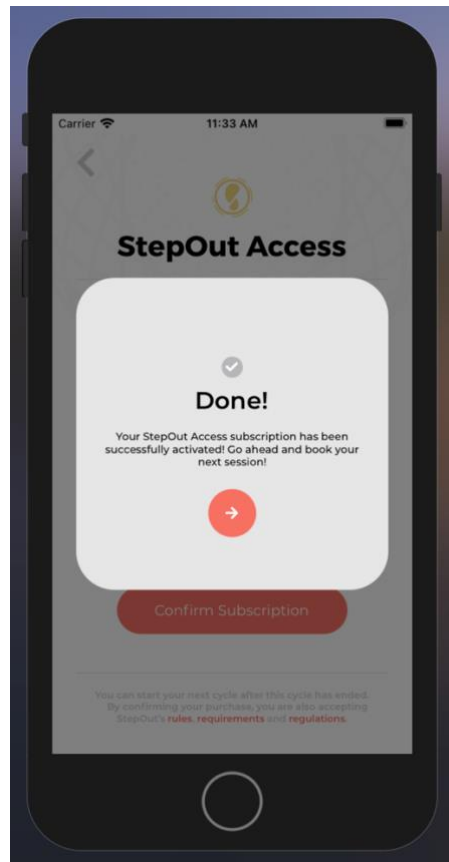


Figure 15. Informative modal box UI when StepOut Access package successfully purchased.

The subsequent tasks were the implementation of business logic functionalities related to StepOut Access package, the development put effort into researching about application data state management within a React Native application, and came to the conclusion of using Redux, a prominent state management library, collaborating with Redux-thunk which is a middleware to support Javascript asynchronous actions handling.

An asynchronous action (function) could be understood as a run-along function that performs its task, however, does not make other functions in the application wait until the asynchronous function is finished. An API call is a perfect example to illustrate this ideology. Usually, an API call will not finish instantly, it requires some

time to receive the response from the backend server, however, the whole application should not be stopped just to wait for some data from that API call. This shows the contrary with Synchronous programming, Javascript was at first - synchronous programming language, in other words, things happen one at a time, if a function that requires a long time to run is called, the program (application) is stopped during the time it takes to perform the action, then continued only every task in that function is finished. /11/

The figure below visualizes the difference between synchronous and asynchronous terminologies. The thick blue lines represent the time frame that the program spends to run normally, costs resource from the hardware; the thin red lines represent time spent by the network to finish receiving what the program needs. It was shown clearly that in an asynchronous model when triggering a network action conceptually causes a split in the timeline, the action happens alongside the running program, notifying the program when it is finished. /11/

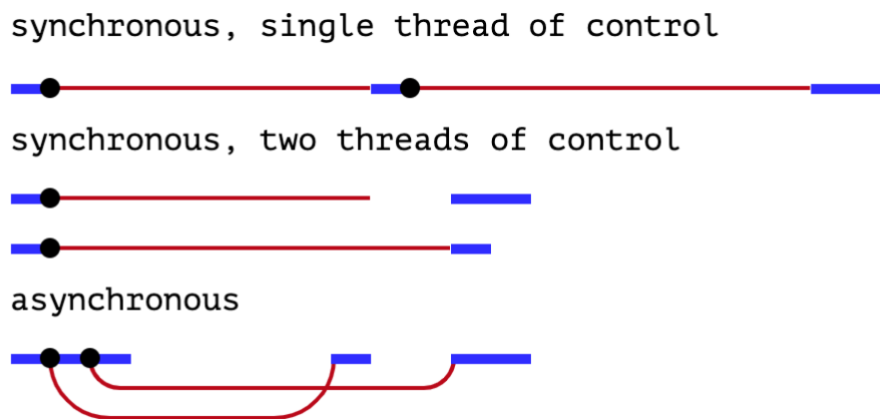


Figure 16. Asynchronous and Synchronous programming. /11/

About Redux, there are core concepts that are essential to understand:

- Redux store: an object that holds every state of the application, the store is the source of data that is available to any component of the app. /12/
- Action creators: functions that return a plain object called action. Action creators are usually invoked when a user interacting with UI (clicking a button, for example); action creators by default are synchronous functions; however, with the assistance from Redux-thunk middleware, action creators can be asynchronous as well. /12/
- Actions: plain objects returned by action creators, have type and payload properties to identify between multiple actions. /12/
- Reducers: pure functions that take the previous global state of the app in Redux store, then depends on the action, return an updated copy of the global state. /12/

The figure below explains the flow in an application handled by Redux

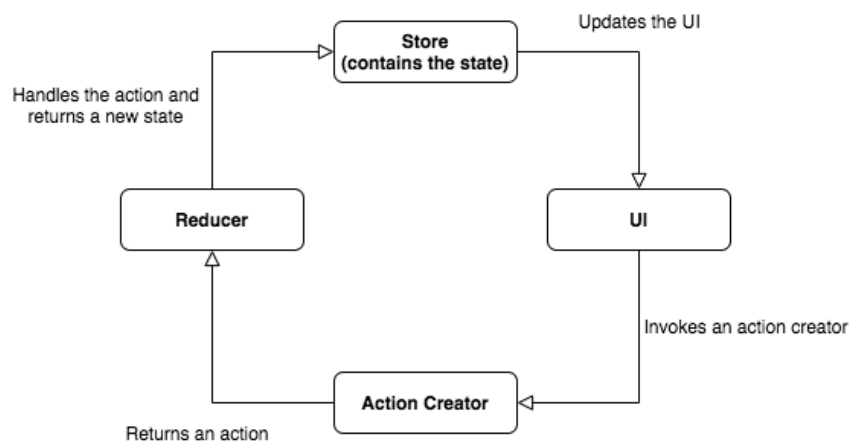


Figure 17. Life cycle of an application with Redux. /12/

After Redux setting up was finished, a considerable amount of working time was used to test out API routes related to StepOut Access subscription package, by sending HTTP requests and handle the response from a server.

The first task using the fetched data retrieved from the server by the GET HTTP method to display not only the information related to the package (price, number of allowance sessions, remaining sessions, start date, and end date) but also to adjust the price when user browsing classes. For example, if an ordinary user who does not have a StepOut Access subscription, when browsing classes, the user will see the actual normal price of that particular class; on the other hand, if they have the subscription package, all the classes which take place within the cycle period will display 0€ price tag. The other task is handling the POST HTTP method to input an access code for activating StepOut Access package.

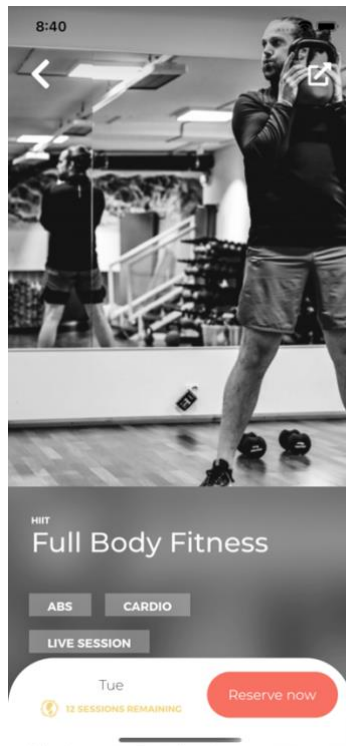


Figure 18. Class Detail when user have StepOut Access.



Figure 19. Purchasing class when having StepOut Access.

After functionalities were handled and tested thoroughly, the application was fine-tuned and made ready for deployments. The new update was reviewed and successfully published on 18 October 2020, meeting the first deadline of the Innovation project.

4.3 Push Notification on Android

A push notification feature to remind users of upcoming events has always been considered as “should have” priority. For an unproblematic solution, integration with Google Firebase was chosen by the StepOut's development team to implement push notifications. Push notification events could be created and scheduled on the Firebase console for showing on end devices that installed the StepOut application.

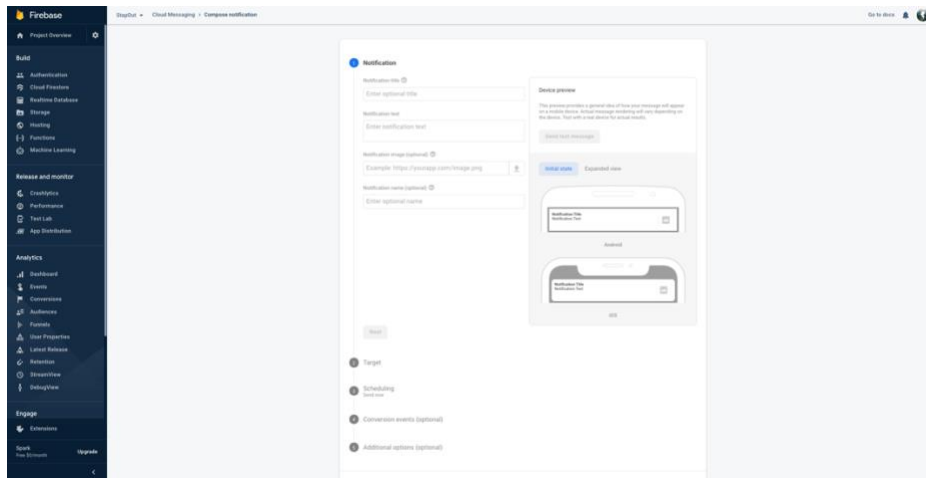


Figure 20. Firebase console UI and push notification creating procedure.

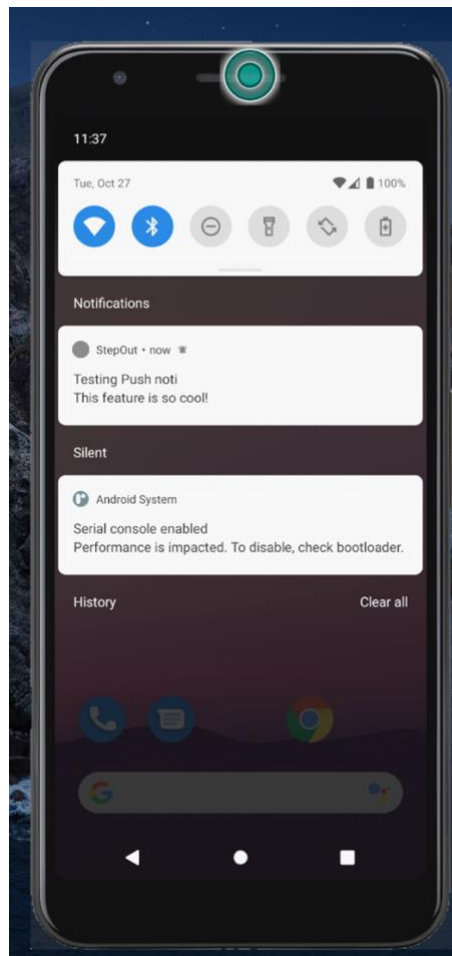


Figure 21. Push notification on Android.

4.4 Implementation of Reward System

The idea of a point system that would benefit users who subscribed with StepOut Access is to earn points based on the number of attended sessions. The earned number of points could be consumed to trade for rewards which are from StepOut's partner: TwoDads, Foodin, and Fuelme; the more valuable the reward is the more points it costs.

The first group of actions for this feature were modifying the StepOut Access intro screen, to introduce a new feature and create more engagement with users, then implementing a brand-new screen – RewardScreen, which contains the detail about the reward and its provider.

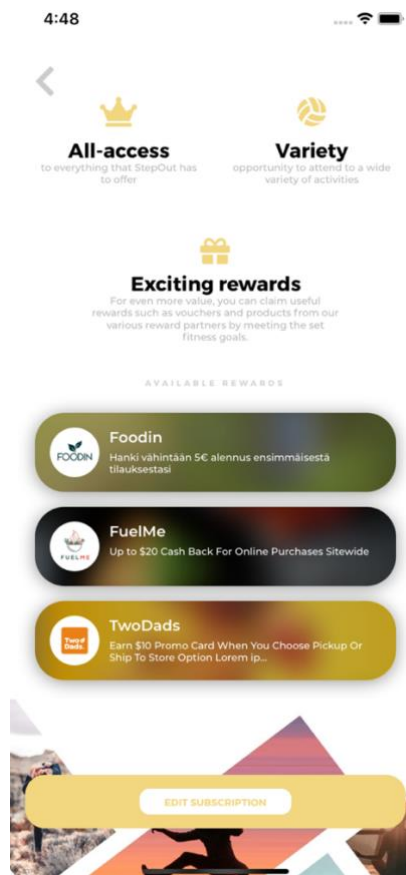


Figure 22. The new appearance of StepOut Access intro screen.

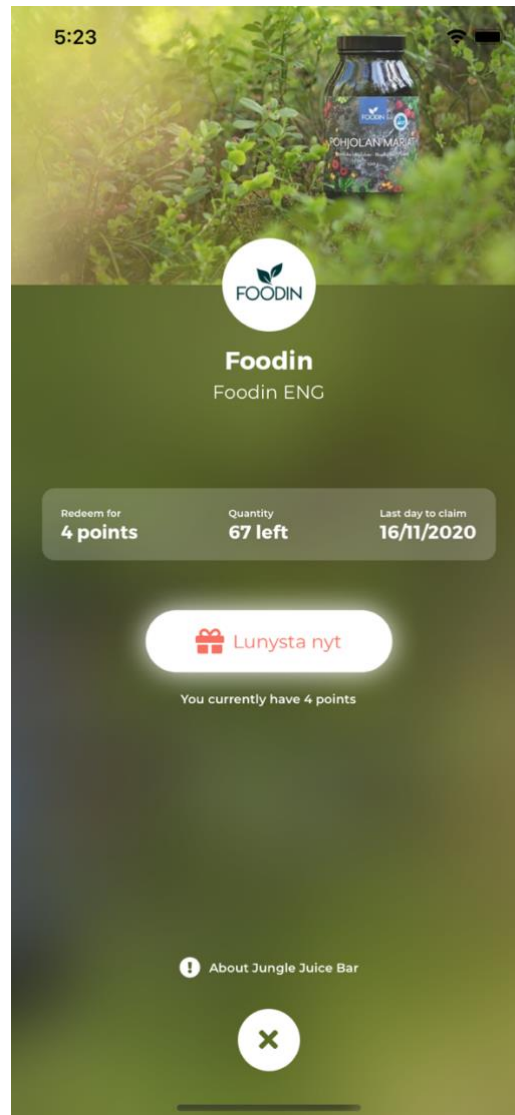


Figure 23. Reward screen implementation.

During the development progress of Reward system, StepOut faced the first obstacle of the MVP since Apple has rejected an update that was meant for fixing a minor issue. The reason for the rejection was Apple demanding their in-app purchase system to be implemented to the application if that application is providing one-to-many real-time experiences /13/; this applies to StepOut's online sessions. In-app purchase implementation means that Apple will have a 30% cut

of every payment made within the application, and as an early-age startup, StepOut will not be able to afford this proportion of revenue loss.

After the issue occurred, a meeting was held for a discussion on the direction of StepOut growing path. Considering a variant of similar study cases, such as Spotify, and Netflix, the StepOut's development team decided to take the same approach as Netflix, since the similarity is translucent, where users would not be able to purchase a subscription with the Netflix mobile application, the only way to activate the subscription is through Netflix website. The meeting ended with a schedule of an improvement of the StepOut website development roadmap, to give the users the ability to purchase the StepOut Access subscription package, which presumably would take place subsequently to the Innovation project with the City of Helsinki completion.

Since then, StepOut has come up with a backup plan of releasing an update with a reward system as a beta-testing build utilize TestFlight testing environment of Apple, therefore, the development process still was proceeded routinely as planted.

Regarding the reward feature, customer private information such as a physical address, name, and phone number, would be a necessary requirement for shipping the rewards. Then, due to the EU GDPR compliance, StepOut would need to collect the user's privacy consent in order to save that kind of data for storing, in that case, an extra crucial feature has been included in the upcoming major update.

First of all, a Policy Update Notification screen will appear whenever there are changes or adjustments in the privacy policy took place.

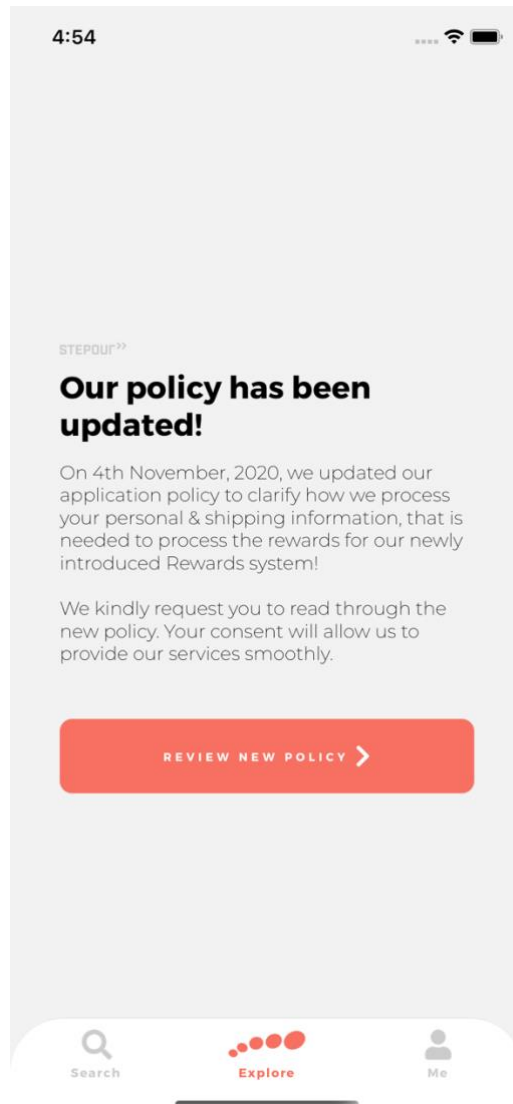


Figure 24. Policy Update Notification screen.

After that, users will be able to review the new policy and make the decision whether to accept those terms and conditions or not; the user will be simply logged out if the consent is declined, otherwise, a pop-up modal will appear to inform the users that the service then could be provided as usual.

Application Policy

Privacy Policy

Thank you for your interest in StepOut
(LetStepOut Oy - 3011634-6)

We at LetStepOut Oy treat the information you submit to us as our own. We only process personal data obtained from you directly when you register for classes or by personally getting in touch with us. This data is processed to register you to the particular classes that you undertake - this includes transferring your data to identify you to our partner trainers for your sign-up confirmation to a class.

The data collected includes:

- Your First Name
- Your Last Name
- Your email address
- Your phone number
- Your profile picture
- Your location

We also process personal information for customer communication purposes, for example when notifying you of changes to training or possible cancellations. Your data is

I ACCEPT

I DECLINE

retention of that data no longer exists. Your location is accessed to determine the time

Figure 25. New policy revision.

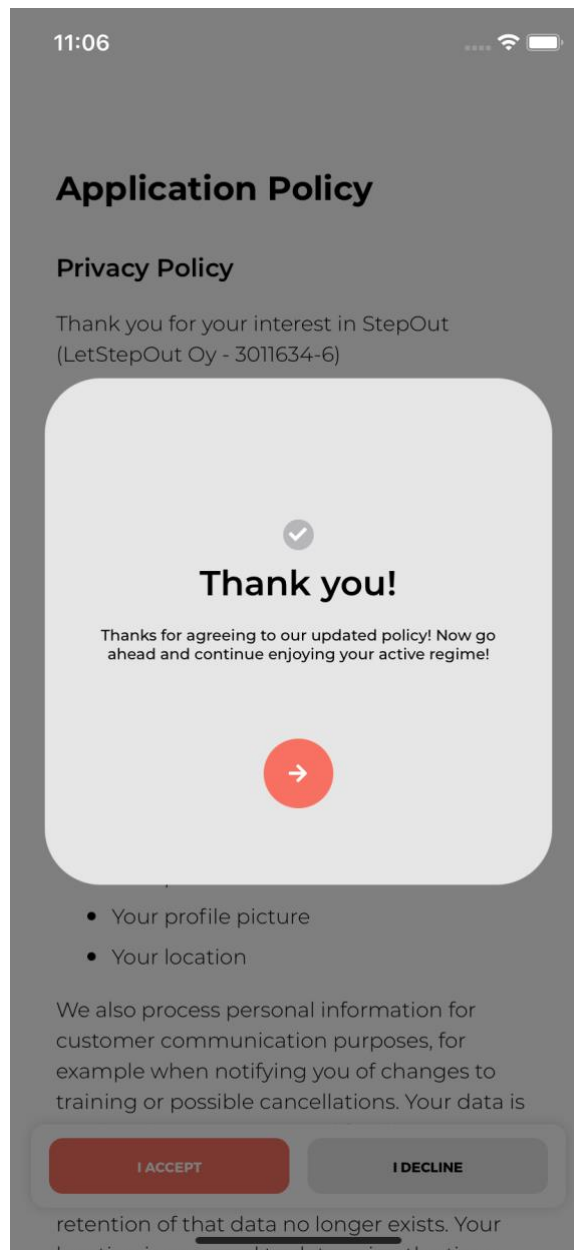
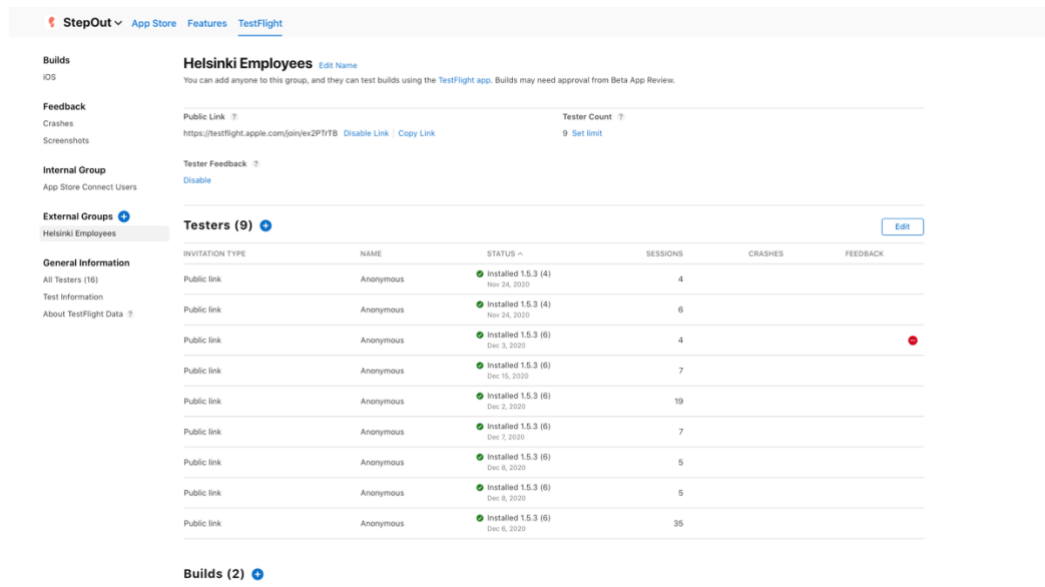


Figure 26. User privacy consent given.

Eventually, the reward feature was completed and was arranged to be released as a beta-testing update the week after.

The new update on Android has been reviewed and approved by Google and everything went smoothly on Android side. Nevertheless, Apple had strict rules

and regulations about one-to-many real-time experiences, therefore, the StepOut new update on iOS has to be released as a beta-testing update for the City of Helsinki users to experiment.



StepOut App Store Features TestFlight

Helsinki Employees [Edit Name](#)

You can add anyone to this group, and they can test builds using the [TestFlight app](#). Builds may need approval from Beta App Review.

Public Link [?](#)
<https://testflight.apple.com/join/ev2P1rTB> [Disable Link](#) [Copy Link](#) Tester Count [?](#)
9 [Set limit](#)

Tester Feedback [?](#)
[Disable](#)

Testers (9) [+](#) [Edit](#)

INVITATION TYPE	NAME	STATUS ^	SESSIONS	CRASHES	FEEDBACK
Public link	Anonymous	● Installed 1.5.3 (4) Nov 24, 2020	4		
Public link	Anonymous	● Installed 1.5.3 (4) Nov 24, 2020	6		
Public link	Anonymous	● Installed 1.5.3 (0) Dec 3, 2020	4		●
Public link	Anonymous	● Installed 1.5.3 (0) Dec 15, 2020	7		
Public link	Anonymous	● Installed 1.5.3 (0) Dec 2, 2020	19		
Public link	Anonymous	● Installed 1.5.3 (0) Dec 7, 2020	7		
Public link	Anonymous	● Installed 1.5.3 (0) Dec 6, 2020	5		
Public link	Anonymous	● Installed 1.5.3 (0) Dec 6, 2020	5		
Public link	Anonymous	● Installed 1.5.3 (0) Dec 6, 2020	35		

Builds (2) [+](#)

Figure 27. Many City of Helsinki users have updated the application via TestFlight.

The remaining tasks were to maintain and support users if any issues occurred, as well as, modifying the email template that each user will receive after booking a session.

Thank you for choosing us!

Your booking for **Moon Flow Yoga** has been confirmed.
Down below is your reservation information.



Yoga

Moon Flow Yoga

With **Weizel Gulfan** as trainer

30th Nov 2020

12:00 pm - 01:00 pm

1 person

0€

Confirmed

Tämä tunti tapahtuu virtuaalisesti.
Vastaanotat tunnin sisäänkirjautumislinkin
kun olet rekisteröitynyt sille.

More info can be found at our [FAQ](#) section.

Thank you for stepping **out** with us!

Love & Power,
Team **StepOut**.

Figure 28. Out-dated template for booking confirmation email.

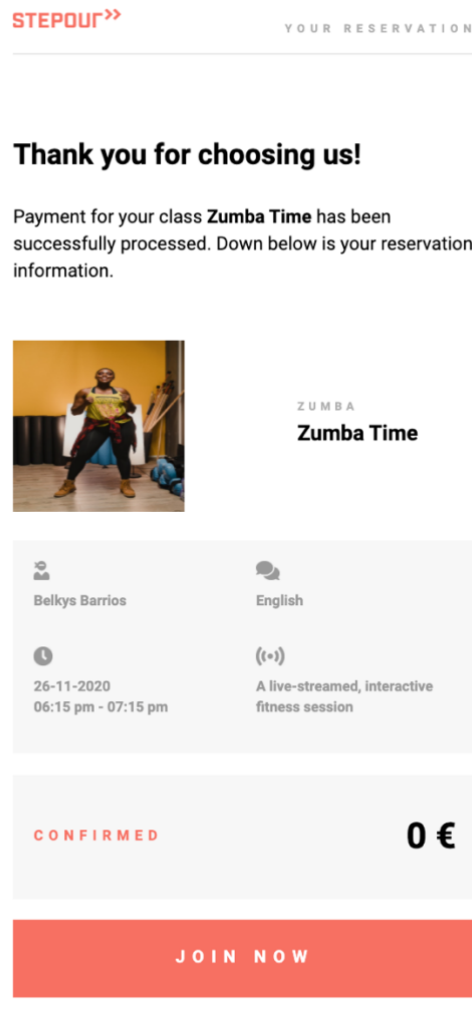


Figure 29. Updated template for booking confirmation email.

4.5 Overview

The innovation project ended with a piece of positive news. Due to the pressure of negative response from a massive number of companies and startups from all over the world, Apple announced that they would extend the deadline for

requiring in-app purchases implementation for applications that providing a one-two-many experience. /14/

Encouraged by this event, the StepOut team prepared to fine-tune the application slightly further and sent out another build for Apple to review.

Another task was to enhance the StepOut website landing page. Since the StepOut Access package has been released and received many positive feedbacks, a decision was made by the StepOut team that it is a suitable time to give StepOut more public exposure.

The remaining tasks included adjustments and functionality improvement for Google login on Android, along with setting up growth hacking channels and advertisements for StepOut service, such as Facebook Ads, Google Analytics, and Hubspot.

5 DEPLOYMENT PROCEDURES

In this section, details about the StepOut application deployment on both Android and iOS platform is demonstrated.

5.1 iOS

For iOS, an Apple developer account is required to set up the deployment to iOS AppStore. Then, a version code of the new build must be generated in order to perform the archiving action. The set of figures below demonstrates the procedure.

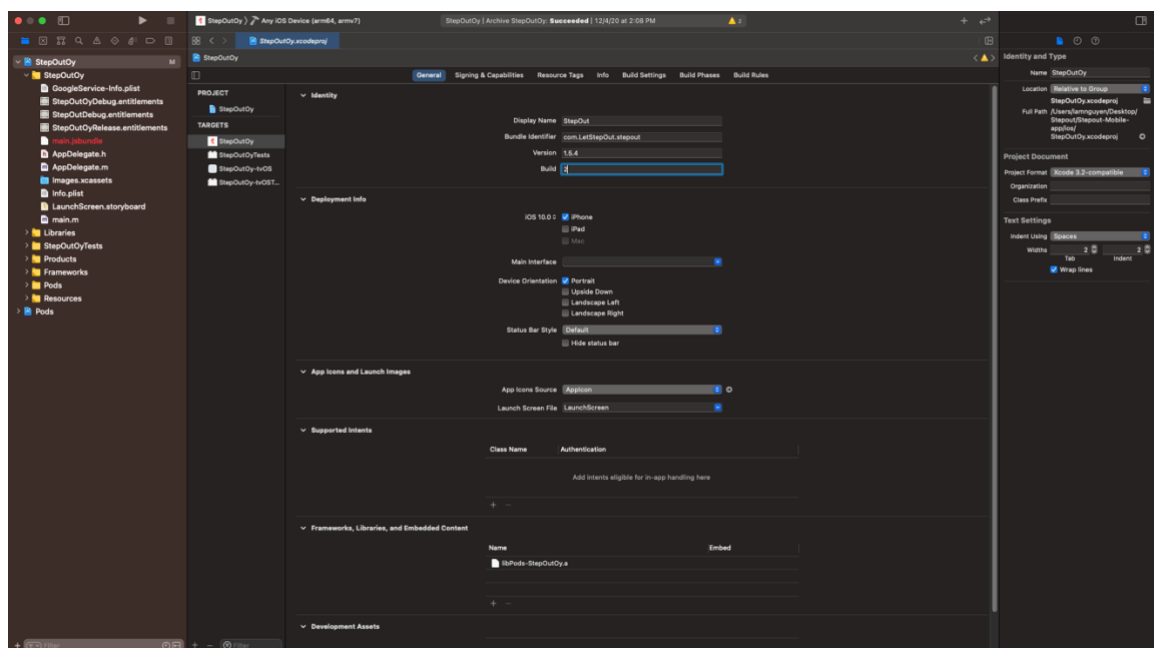


Figure 30. General configurations for iOS project.

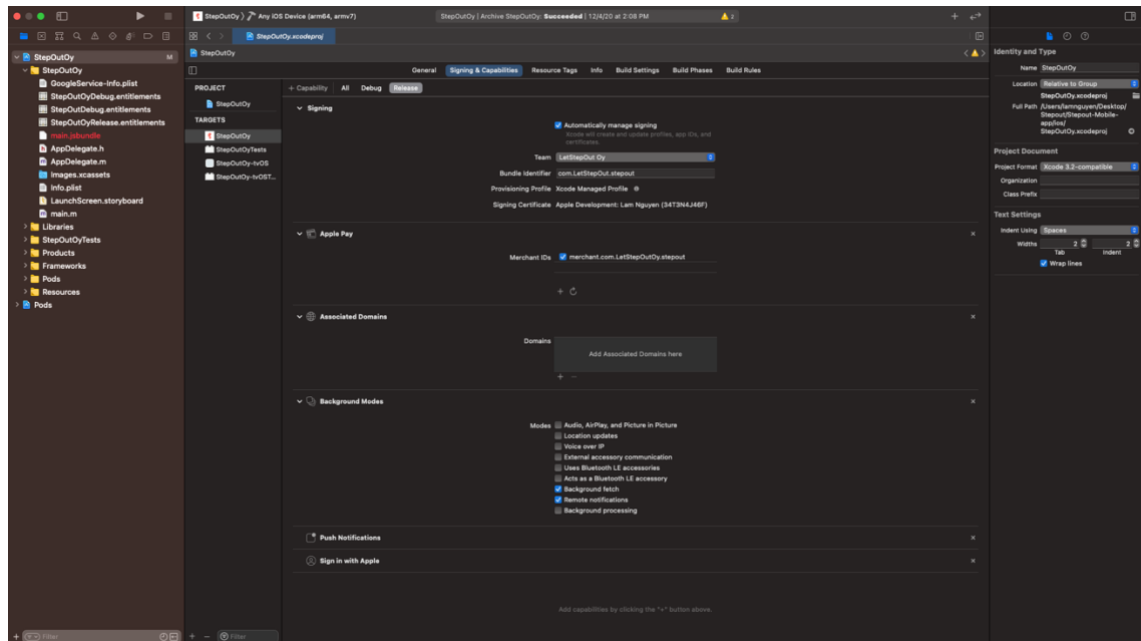


Figure 31. Apple developer certificates and application capability configurations.

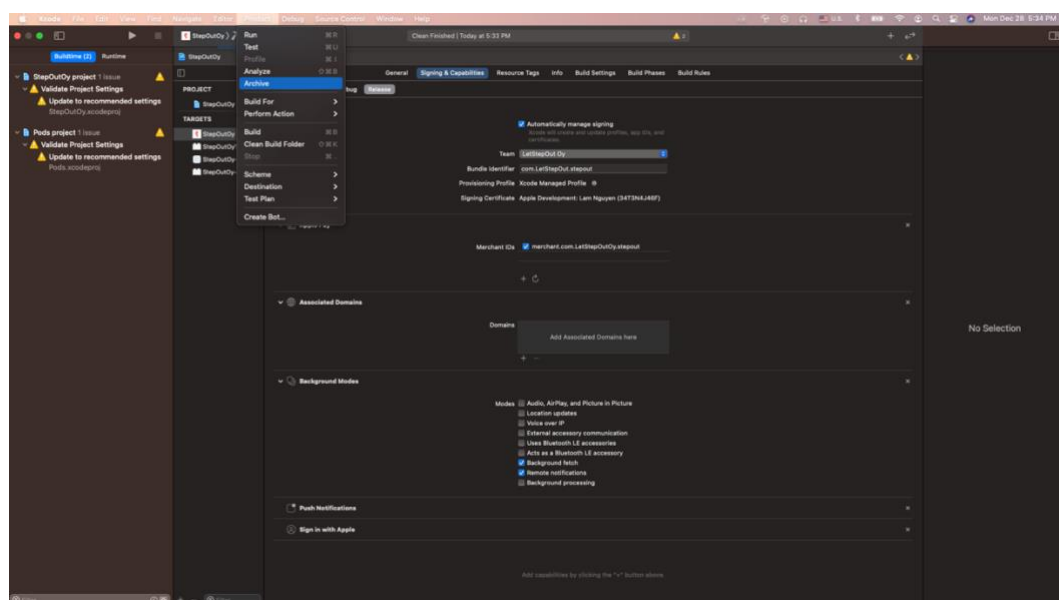


Figure 32. iOS build action for archiving.

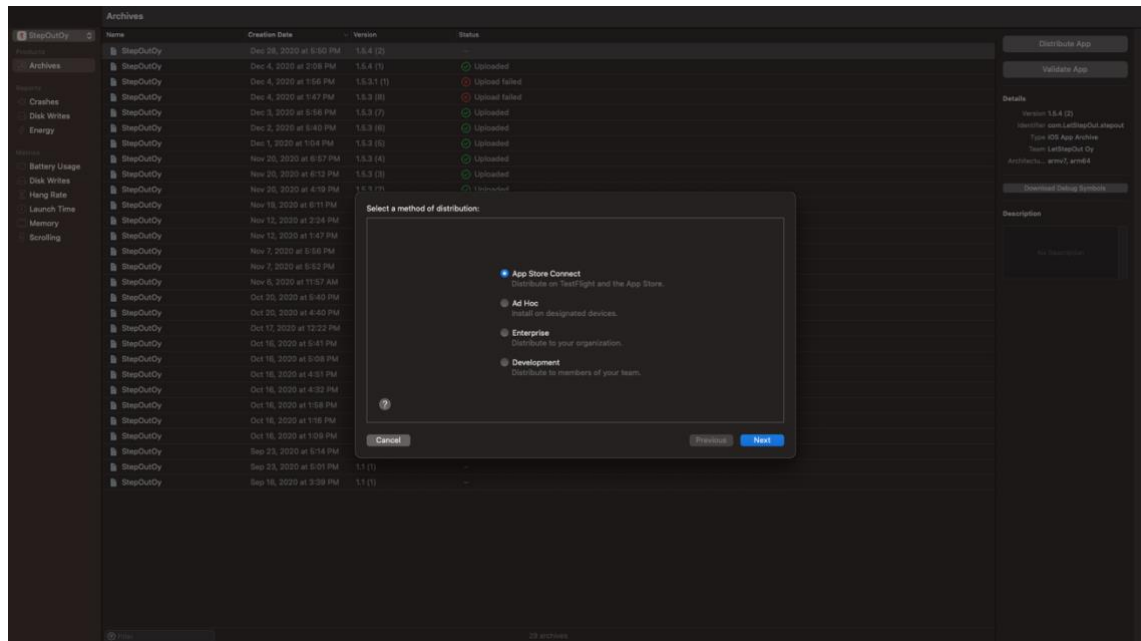


Figure 33. Distribution (deployment) options.

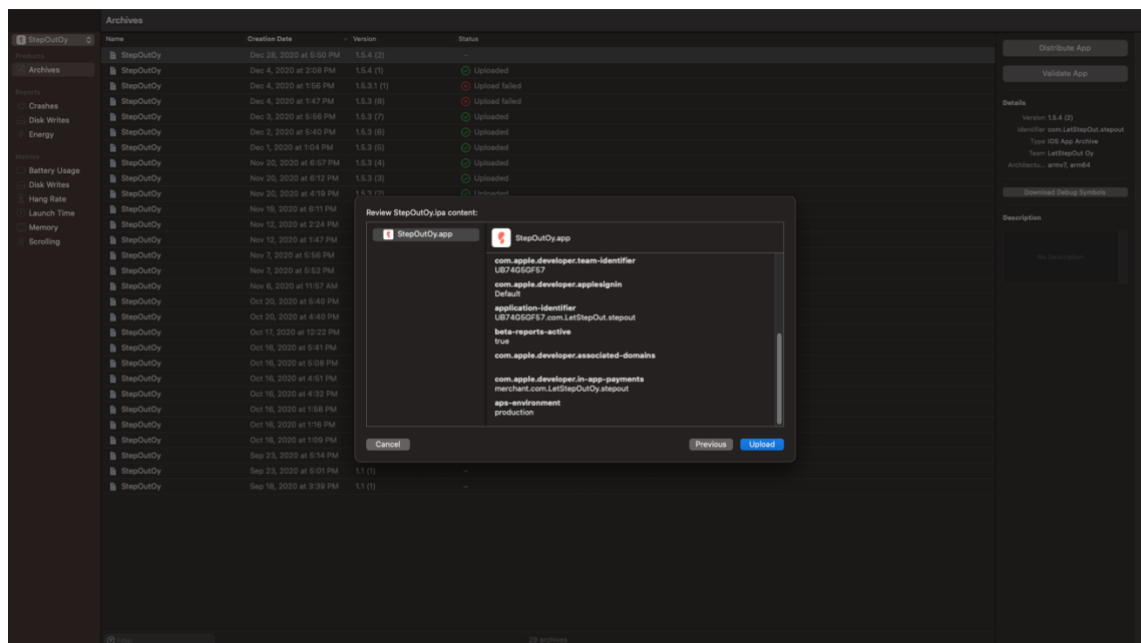


Figure 34. Apple distribution certificate signed and ready to upload.

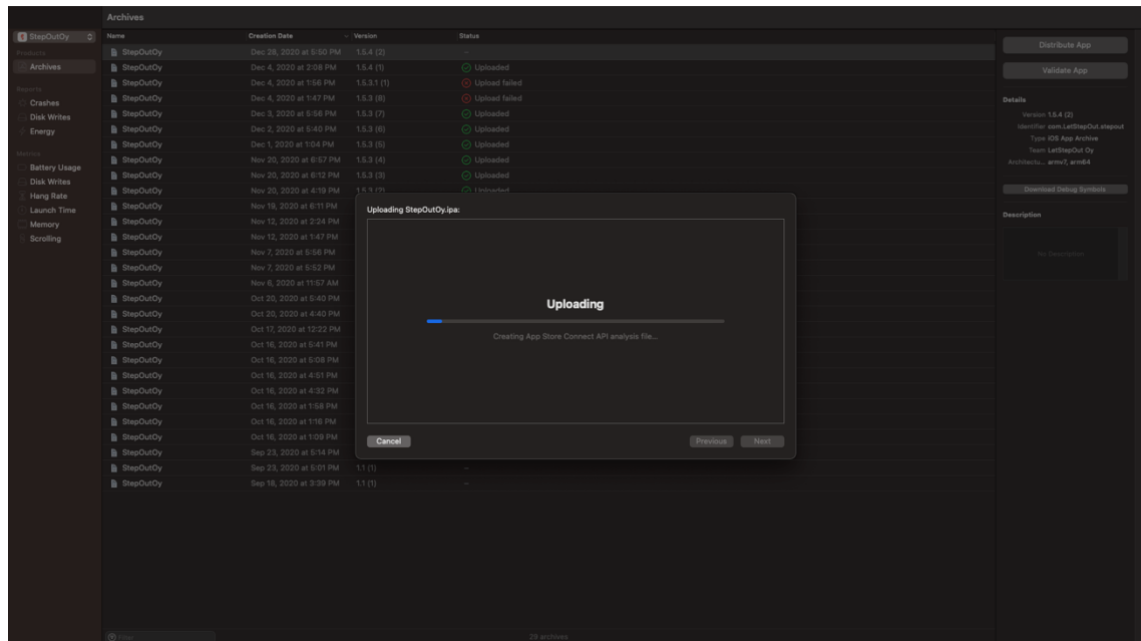


Figure 35. iOS distribution has been initiated.

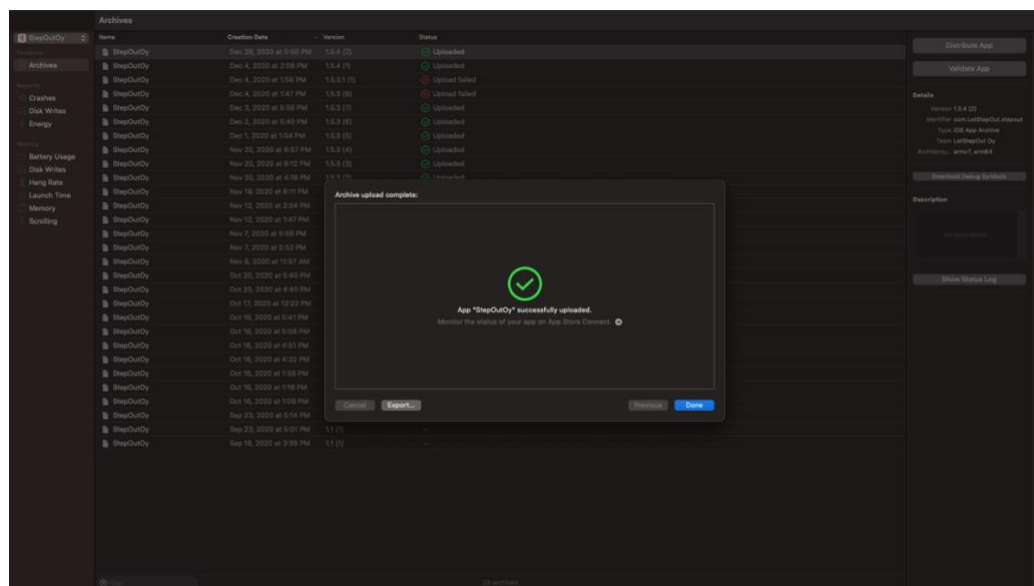


Figure 36. Build has been uploaded to Apple Connect successfully.

Once the step in Figure 36 has been reached, thus far it is not completed for the new update available for all users on iOS AppStore. Apple requires an additional action for publishing the update on AppStore, which is submitting the new build for Apple for reviewing, in order to perform an assessment of the new build's compliance with Apple regulations and rules.

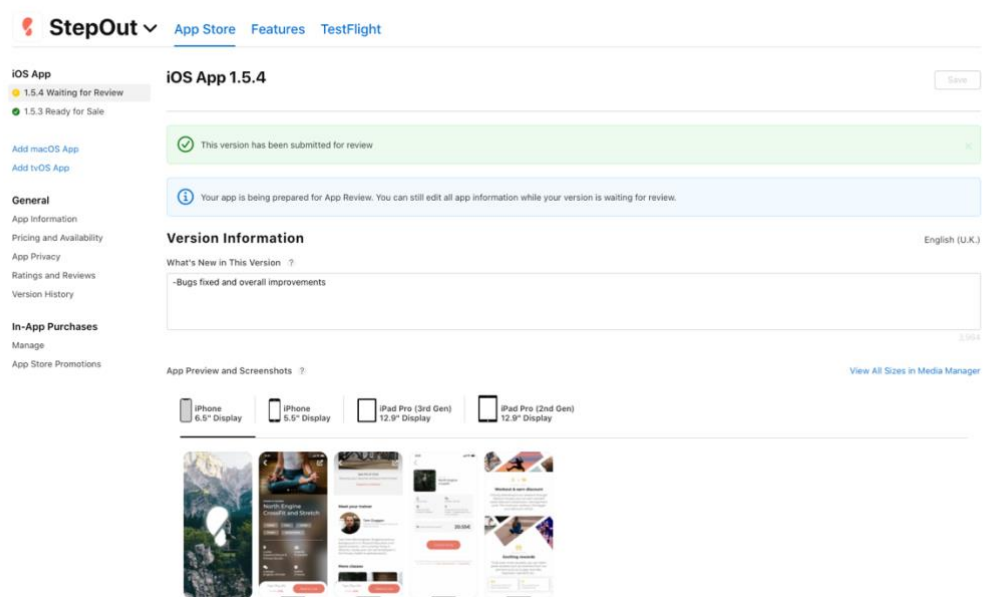


Figure 37. iOS new update submission for Apple review.

After the review is processed and approved, the new update will be available for every user to download on iOS platform.

5.2 Android

On the Android side, the deployment procedure is considerably comparable, where it comes down to an equivalent pattern: new build generation -> importing/uploading the build to Google Play console -> build submission for Google to review -> new update published on Google Play.

The set of figures provides an enhanced understanding of the deployment procedure from the Android platform point of view.

Firstly, a version code and version name must be specified for the new build in Android project “build.gradle” file.

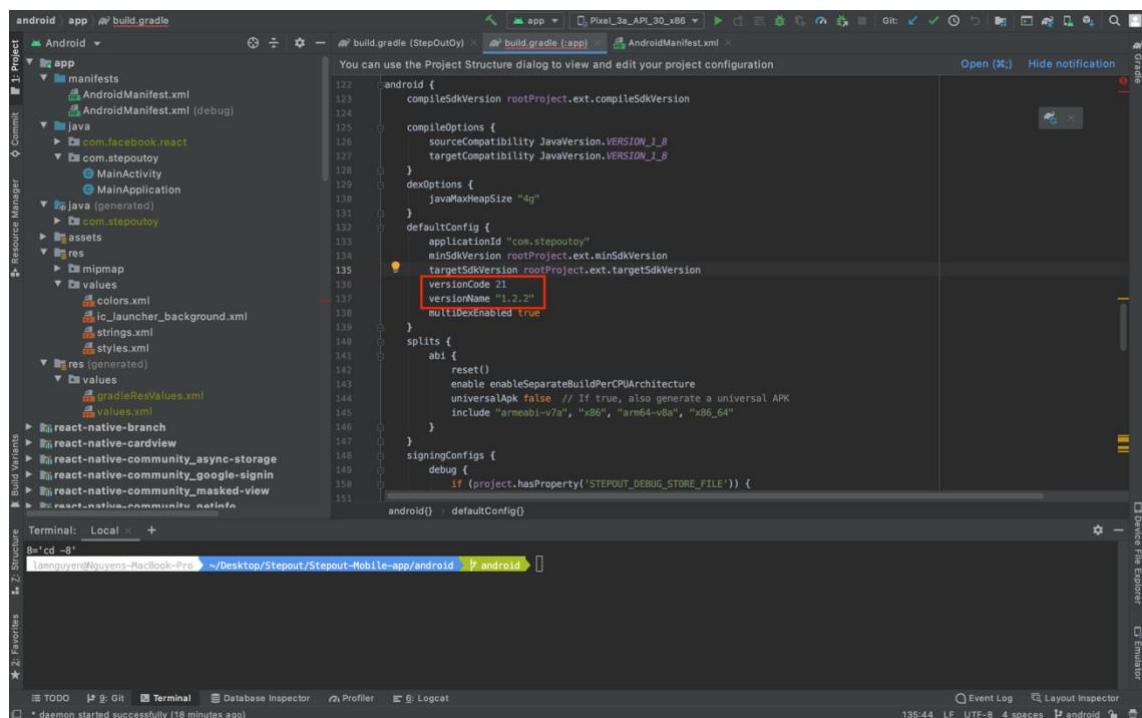


Figure 38. Android version modification.

The next step is running “./gradlew app:assembleRelease” in the Android project directory command to build the release APK file.

```

Terminal: Local +
B="cd -B"
lamnguyen@Nguyens-MacBook-Pro: ~/Desktop/Stepout/Stepout-Mobile-app/android → android ./gradlew app:assembleRelease
Starting a Gradle Daemon (subsequent builds will be faster)

> Configure project :react-native-firebase_app
:react-native-firebase_app package.json found at /Users/lamnguyen/Desktop/Stepout/Stepout-Mobile-app/node_modules/@react-native-firebase/app/package.json
:react-native-firebase_app:firebase.bom using default value: 25.12.0
:react-native-firebase_app:play-services-auth using default value: 18.1.0
:react-native-firebase_app package.json found at /Users/lamnguyen/Desktop/Stepout/Stepout-Mobile-app/node_modules/@react-native-firebase/app/package.json
:react-native-firebase_app:version set from package.json: 8.4.6 (8,4,6 - 8004006)
:react-native-firebase_app:android.compileSdk using custom value: 29
:react-native-firebase_app:android.targetSdk using custom value: 29
:react-native-firebase_app:android.minSdk using custom value: 16
:react-native-firebase_app:reactNativeAndroidDir /Users/lamnguyen/Desktop/Stepout/Stepout-Mobile-app/node_modules/react-native/android

> Configure project :react-native-firebase_messaging
:react-native-firebase_messaging package.json found at /Users/lamnguyen/Desktop/Stepout/Stepout-Mobile-app/node_modules/@react-native-firebase/messaging/package.json
:react-native-firebase_app package.json found at /Users/lamnguyen/Desktop/Stepout/Stepout-Mobile-app/node_modules/@react-native-firebase/app/package.json
:react-native-firebase_messaging:firebase.bom using default value: 25.12.0
:react-native-firebase_messaging package.json found at /Users/lamnguyen/Desktop/Stepout/Stepout-Mobile-app/node_modules/@react-native-firebase/messaging/package.json
:react-native-firebase_messaging:version set from package.json: 7.9.1 (7,9,1 - 7009001)
:react-native-firebase_messaging:android.compileSdk using custom value: 29
:react-native-firebase_messaging:android.targetSdk using custom value: 29
:react-native-firebase_messaging:android.minSdk using custom value: 16
:react-native-firebase_messaging:reactNativeAndroidDir /Users/lamnguyen/Desktop/Stepout/Stepout-Mobile-app/node_modules/react-native/android

> Configure project :react-native-illn
WARNING: Configuration 'compile' is obsolete and has been replaced with 'implementation' and 'api'.
It will be removed in version 5.0 of the Android Gradle plugin.
For more information, see http://d.android.com/r/tools/update-dependency-configurations.html.

> Configure project :react-native-timzone
WARNING: Configuration 'compile' is obsolete and has been replaced with 'implementation' and 'api'.
It will be removed in version 5.0 of the Android Gradle plugin.
For more information, see http://d.android.com/r/tools/update-dependency-configurations.html.

> Task :app:bundleReleaseJsAndAssets
warning: the transform cache was reset.
Welcome to React Native!
Learn once, write anywhere

```

Figure 39. Execute “./gradlew app:assembleRelease” for building a release APK file.

The generated APK will be located at “./android/app/build/outputs/apk/release” path.

```

Terminal: Local x +
lamnguyen@Nguyens-MacBook-Pro: ~/Desktop/Stepout/Stepout-Mobile-app/android/app/build/outputs/apk/release → android ± pwd
/Users/lamnguyen/Desktop/Stepout/Stepout-Mobile-app/android/app/build/outputs/apk/release
lamnguyen@Nguyens-MacBook-Pro: ~/Desktop/Stepout/Stepout-Mobile-app/android/app/build/outputs/apk/release → android ± ls -la
total 90008
drwxr-xr-x  4 lamnguyen  staff   128 Dec 30 13:50 .
drwxr-xr-x  5 lamnguyen  staff   160 Dec 30 13:55 ..
-rw-r--r--@ 1 lamnguyen  staff 45065431 Dec 30 13:50 app-release.apk
-rw-r--r--  1 lamnguyen  staff   337 Dec 30 13:50 output-metadata.json
lamnguyen@Nguyens-MacBook-Pro: ~/Desktop/Stepout/Stepout-Mobile-app/android/app/build/outputs/apk/release → android ±

```

Figure 40. Locating generated APK file.

The APK file then would be submitted for Google to review and only published when complying with Google regulations and rules.

Release dashboard [Releases](#) Countries/regions

Releases

21 (1.2.2)
✔ Available on Google Play · Last updated 15 Dec 16:10 · Available in 151 countries · Available on 15,687 devices
[View release details](#)

Release history

Hide ^

Release	Released	Status	
19 (1.2.1)	2 December 2020	Replaced on 15 Dec 16:10	→
16 (1.2)	20 November 2020	Replaced on 2 Dec 13:02	→
13 (1.1.2)	6 November 2020	Replaced on 20 Nov 19:53	→
12 (1.1.1)	20 October 2020	Replaced on 6 Nov 19:15	→
10 (1.1)	17 October 2020	Replaced on 20 Oct 20:15	→
1.0	26 August 2020	Replaced on 17 Oct 12:59	→

Figure 41. An approved production-ready build on Google Play.

6 CONCLUSIONS

Overall, the new features of the StepOut mobile application were accomplished for the most part. The StepOut mobile application was developed and powered by React Native, an innovative technology, and the features have been achieved precisely so far. However, it is undeniable that the technology itself is still lacking some stability and compatibility. For instance, the StepOut development team was faced with difficulties when implementing Push Notification on the iOS platform due to Firebase failing the integration to React Native. Despite of the effort of following the Firebase instruction and further configuration on the iOS native layer, the push notification has never worked on iOS.

The development process was a learning curve that the author has taken, alongside with the support from other team members and developers. Working with React Native was a fascinating experience though, challenging at the same time; lack of mobile development experience at first challenged the author with any native configuration and feature implementation. However, React Native was powered by React library and it gave the author the confidence to get over the learning curve over time since the author was familiar with website development and Javascript frameworks overall. Another struggle the author has faced during the deployment procedure was figuring out the structure of development certificates and the “provisioning profile” system required by Apple to be able to push update to App Store.

Furthermore, as an observation, it could be said that cross-platform mobile development is beneficial when taking financial and time frame into consideration, on the other hand, when considering maintainability and stability, Native mobile development is always a better choice.

The growth path of the StepOut mobile application is far from done, in the future, a new payment handling system (could possibly be related to in-app-purchase) for both iOS and Android should be implemented properly. However, the current version of the StepOut application is still having some minor performance issues, as well as inconsistent UI elements which could be maintained and fine-tuned in the long run.

REFERENCES

- /1/ Europe Active. Out now: the European Health & Fitness Market Report 2020. Accessed 8 January 2021. <https://www.europeactive.eu/news/out-now-european-health-fitness-market-report-2020>
- /2/ Technogym. How and where fitness market is growing in Europe. Accessed 8 January 2021. <https://www.technogym.com/int/newsroom/fitness-market-growth-europe/>
- /3/ StepOut website. About Us. Accessed 8 January 2021. <https://www.stepout.fi/ole-hyva>
- /4/ Netguru. What is React Native and How it is used? Accessed 8 January 2021. <https://www.netguru.com/what-is-react-native>
- /5/ Hackernoon. Native Anroid or iOS vs React Native: Which one to choose? Accessed 8 January 2021. <https://hackernoon.com/native-android-or-ios-vs-react-native-which-one-to-choose-jg3131y1>
- /6/ Devathon. How native is React Native? React Native vs Native App development. Accessed 8 January 2021. <https://medium.com/@devathon /how-native-is-react-native-react-native-vs-native-app-development-ee65c4ba642b>
- /7/ Pawel & Karol Wegner. 13 April 2020 – Pros and Cons of React Native and Native Development. Accessed 8 January 2021. <https://itcraftapps.com/blog/pros-and-cons-of-react-native-and-native-development/>

- /8/ Dileep, Gupta. 28 December 2018 – React Native vs Native: What to choose for App Development. Accessed 8 January 2021. <https://appinventiv.com/blog/react-native-vs-native-apps/>
- /9/ Scott, Lowe. 27 January 2015 – Using the Fork-and-Branch Git Workflow. Accessed 8 January 2021. <https://blog.scottlowe.org/2015/01/27/using-fork-branch-git-workflow/>
- /10/ Stripe. Strong Customer Authentication. Accessed 8 January 2021. <https://stripe.com/en-fi/guides/strong-customer-authentication>
- /11/ Marijn, Haverbeke. 2018. Eloquent Javascript (3rd Edition) – Asynchronous Programming. Accessed 8 January 2021. https://eloquentjavascript.net/11_async.html
- /12/ Tyler, Hawkins. 15 February 2020 – Understanding the Flow of a React + Redux Application. Accessed 8 January 2021. <https://medium.com/better-programming/understanding-the-flow-of-a-react-redux-application-4dc8c8da08c7>
- /13/ Apple Inc. Apple Store review guidelines. Accessed 8 January 2021. <https://developer.apple.com/app-store/review/guidelines/ - person-to-person-experiences>
- /14/ Apple Inc. Deadline extended for online group event in-app purchase requirement. Accessed 8 January 2021. <https://developer.apple.com/news/?id=kw16tplo>
- /15/ StepOut blog page. StepOut X Helsingin Kaupunki. Accessed 8 January 2021. <https://stepout.fi/blog/2020/11/17/stepout-x-helsingin-kaupunki/>