LAPIN AMK
Lapland University of Applied Sciences

DATABASE DESIGN FOR THE ANDROID PLATFORM

Memory Game Database Design

Aina Oluwasegun

Bachelor's thesis
School of Business and Culture
Degree Programme in Business Information Technology
Bachelor of Business Administration

2016

LAPIN AMK
Lapland University of Applied Sciences

Abstract of Thesis

School of Business and Culture
Business Information Technology
Business Administration

Author Aina Oluwasegun Year 2016
Supervisor Johanna Vuokila
Commissioned by
Title of Thesis Database Design for the Android Platform – Memory Game Database Design
No. of pages + app. 40 + 4

There are thousands of mobile applications existing in a market that seems to be growing rapidly as the information technology sector develops. These applications vary in the software used for the development, data size and the design to fit their purpose for example in the entertainment, medical, business and finance sector of the economy and the society at large.

The commissioning company Datero Oy requested for a game that will be published for Android platform. Presently, the implementation of the game exists, but it cannot be published because it has several hitches.

The aim of this thesis work is to design the database of a mobile game application i.e. Memory Game for the Android platform.

The qualitative research approach was one of the research approaches used in this thesis work. Based on the qualitative research approach some data was collected through an interview with the supervisor and some literature was studied.

The constructive research method is used to evaluate the previous game and provide a solution for the new game. Some of the solutions proffer for the new game is achieving data storage efficiency and a bigger database due to the fact that picture and sound features will be added to the new memory game.

The result of this thesis work is designing a database with SQLite language for the Android Platform for better processing of data. This is derived from the findings and the literature review on the design of the database.

The assignor and other members of the team, that are part of designing the memory game welcomed this suggestion because SQLite is compatible with other software that will be used by other members of the team in the cause of design and development of the memory game at a later day.

Key words Database, Android, SQLite, MySQL, Memory Game

CONTENTS

SYMBOLS AND ABBREVIATIONS

ASE	Sybase Adaptive Server Enterprise
IDI	Individual in-depth Interview
DBMS	Database Management System
ANSI	American National Standards Institute
SPARC	Standards Planning and Requirements Committee
DDLC	Database Development Life Cycle
SDLC	Software Development Life Cycle
ERD	Entity Relationship Diagram
DDL	Data Definition Language
RDBMS	Relational Database Management System
NoSQL	Not Only SQL
WVGA	Wide Video Graphics Array
QVGA	Quarter Video Graphics Array
GC	Garbage Collector
API	Application Programming Interface
JIT	Just-in-time
VM	Virtual Machine

FIGURES AND TABLES

1 INTRODUCTION

This chapter discusses the background and motivation of this thesis work, with the brief description of the commissioning company. Furthermore, the objectives of the research are described and the structure of the thesis outlined.

1.1 Background and Motivation

The commissioning company Datero Oy, offers information and communication technology for a group of people with special needs. They work on technical reading and writing aids, software and application that will help this group of people. The commissioning requested for a memory game that will be published on the Android platform. Presently, the implementation of the game exists but it cannot be published because it has some hitches, which are some functions in the game are not working properly. On these bases, the database for the new game will be integrated into the game, for better processing of the data.

The Android platform has been one fastest growing platforms as compared to other platforms. For this reason, most of the game developers have chosen this platform instead of other platforms. Gargenta (2011, 1) was also of the opinion that Android is transforming the mobile space, as an open platform that separates the hardware from the software that operates on it.

The commissioning company Datero Oy, requested for a game that will help train the brain for persons with special needs. Datero Oy has been in the service of providing a workshop, training and games for persons with special needs as a means of giving back to the society.

My interest in MySQL during the cause of my study and the zeal to learn more on different database language used for the development of database has been my motivations for this thesis. I might also consider a database analyst as a profession later in the future.

1.2 Scope and Objectives

This research work focuses on designing a database of a Memory Game on SDLC framework and the database language used for the databases is SQLite. SQLite is a database language which is compatible with the Android platform. The goal of the game is to train the brain to memorize by pairing similar pictures on the screen correctly.

The objective of the thesis study is to design a database with SQLite language for the Android platform. Furthermore, new features are added to the new game by improving the functionalities, thereby expanding the database by creating more classes. By so doing, the game becomes more interesting to the end user than the previous game.

The game design is divided into three major parts as follows: interface, software and databases design with each individual of the team of three focusing on one part of the design. The part of the game design reported in this thesis focuses on database design and highlights the major aspect of the design process.

1.3 Structure of the Thesis

This Thesis is divided into six chapters. Research questions and methodology are discussed in chapter 2. Database environments, management system, development and relational database are discussed in chapter 3. Introduction and versions of Android are discussed in chapter 4. The Introduction of SQLite, as an embedded database, and users of SQLite are discussed in chapter 5. Chapter 6 draws the conclusion of the thesis work.

2 RESEARCH QUESTIONS AND METHODOLOGY

This chapter discusses the research questions and methodologies used for the research work.

2.1 Research Questions

Base on the interview with the supervisor on the design of the former game and the perception and recommendations of the commissioning company on what needs to be done in the new game design. In order to align the design of the database to meet the recommendation of the company, some questions need to be answered. Answering these questions will be a guide to achieving the solution to the existing problems.

1. How does database design framework affect the accessibility of data on a mobile application for Android platform?

Before deciding on the type of design framework to use for a database, there must be a clear knowledge of the type application the database will serve. A good understanding will propel a well-organized design framework with a high degree of success.

2. Does SQLite serve as an alternative to SQL server for Android platform?

Though SQLite support is inbuilt in Android, it is not the only solitary means to data sustenance for applications. Some other database system can be used, for example, JavaDB or MongoDB but the application and the needed libraries will be put together because they can't depend on Android's built-in database assistance. SQLite is not a substitute to SQL server but the substitute to local file usage on the illogical format. (Gargenta 2011,120.)

2.2 Methodology

Denzin and Lincoln (1994 as cited by Jha 2008, 45) recognized that qualitative research is perceived by different people in different ways. This is seen as a generic definition.

The qualitative research approach was selected for this thesis work base on the interview with the supervisor. The qualitative research is multi-dimensional in focus, relating to interpretive, naturalistic way to the topic in focus. On this assertion, qualitative research can be said to study things in their normal way. Qualitative research revolves around the studies use to gather various empirical materials. For example interview, personal experience, case study, life story, interactions and challenging moments. (Jha 2008, 45.)

The interview is a major data gathering method in qualitative methodologies. There are different types of interviews, and this depends on the number of people that are engaged in the interview and the level of formation.

An interview can be performed individually, individual in-depth interview (hereinafter IDI) or as a group. The table 1 below as extracted from (Sachdeva, 2009, 168) distinguishes individual and group interview as a research methodology.

Table 1. Differences between individual and group interview as a research methodology

Sachdeva (2009, 168)

Individual Interview	Group Interview
Research Objectives **.** Explores the individual in depth. Creates case histories through repeated interviews overtime .Test a survey	**.** Orient the researcher to a field of inquiry and language of field. . Explores a range of altitude, opinions and behavior . Add contextual detail to quantitative findings
Topic Concerns **.** Detailed individual experiences, choices ,biographies . Sensitive issues that might provoke anxiety	**.** Issues of public interest . Issues where little is known or of hypothetical nature
Participants **.** High status or elites . Those who have good language skills . Participants whose distinction would inhibit participation	**.** Participants whose backgrounds are similar or not so dissimilar as to generate conflict or discomfort. People can articulate the ideas . People who offer a range of positions on issues

The Constructive research approach was adopted for this thesis work, as one of the suitable approaches for providing a solution to the new game. According to Dodig-Crnkovic (2010), Constructive research approach entails construction of an artifact (practical, theoretical or both) that explain how a particular problem can be solved, by giving clear understanding (knowledge, modeled or explained) on the processes on of solving the problem.

Dodig-Crnkovic (2010) further explained Constructive research provides a result which can be both practical and theoretical relevance. The research should provide the solution to most of the related knowledge problem, concerning practicability, improvements and innovation. More priority should be on the theoretical importance of the construct. Lukka (2001) was also of the opinion “constructive approach is innovative construction productive methodology, which aims to solve a real-world problem, and is in this ways it generates contributions for the scientific field in which it is applied”. Figure 1 below illustrates the major elements of constructive research methodology.

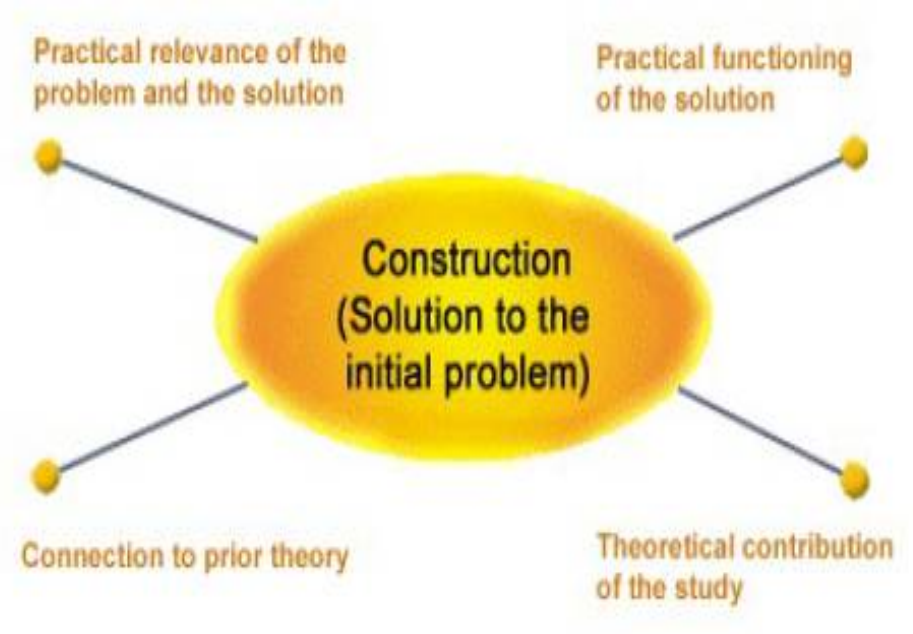


Figure 1. Major Elements of constructive research methodology (Lukka 2001)

Lukka (2001) further refers to constructive research approach as illustrated in the figure above as a pragmatist philosophy which was derived from practical analysis of work to be done that can bring about significant theoretical contributions to relief.

3 DATABASE DESIGN

3.1 Introduction to Database Environments

There is no single definition for a database, and different authors or software vendors give their definitions or perspectives of what a database is. Oppel (2009, 4) define a database as a compilation of data controlled by a single entity. However, Powell (2006, 1) states a database is the collection of information that is correlative related and organized.

These definitions are extensive because there are so many types of databases that exist across the different vendors that provide the databases system. For example, Microsoft Access stores the whole database solitarily in a data file, an Access database is defined as the file that controls data items. Oracle Corporation as cited by Oppel (2009, 4) also defines it database as the compilation of substantial files that are controlled by a database software product. Microsoft SQL Server and Sybase Adaptive Server Enterprise as cited by Oppel (2009, 4) also maintains that a database is a compilation of data entry that has a general owner, and multiple databases are usually controlled by database management software. These definitions can be confusing when working with different products. For example, a database defined by Microsoft SQL Server or Sybase is known by Oracle Corporation as schema (Oppel 2009, 4).

3.2 Database Management System

Database management system (hereinafter DBMS) is software that allows users to define, create, maintain, and control access to the database. Murthy (2008, 8) states the database management systems operate as a supplementary layer of software connecting the operating system and the application program. However, there are different types of layers in database management systems. Including External layer, Logical layer, and the Physical layer.

Oppel (2009, 25) refers to these layers as abstraction layers, because it makes it possible for different data users to view the same database patterned to each user needs. More so, views store no precise data, but they repeatedly produce any data modification made on the basic database objects. The illustration shown in Figure 2 is the architecture of the abstraction layers. It was first developed by American National Standards Institute/Standards Planning and Requirements Committee (hereinafter ANSI/SPARC). The architecture shown below in Figure 2 was adopted from Oppel (2009).

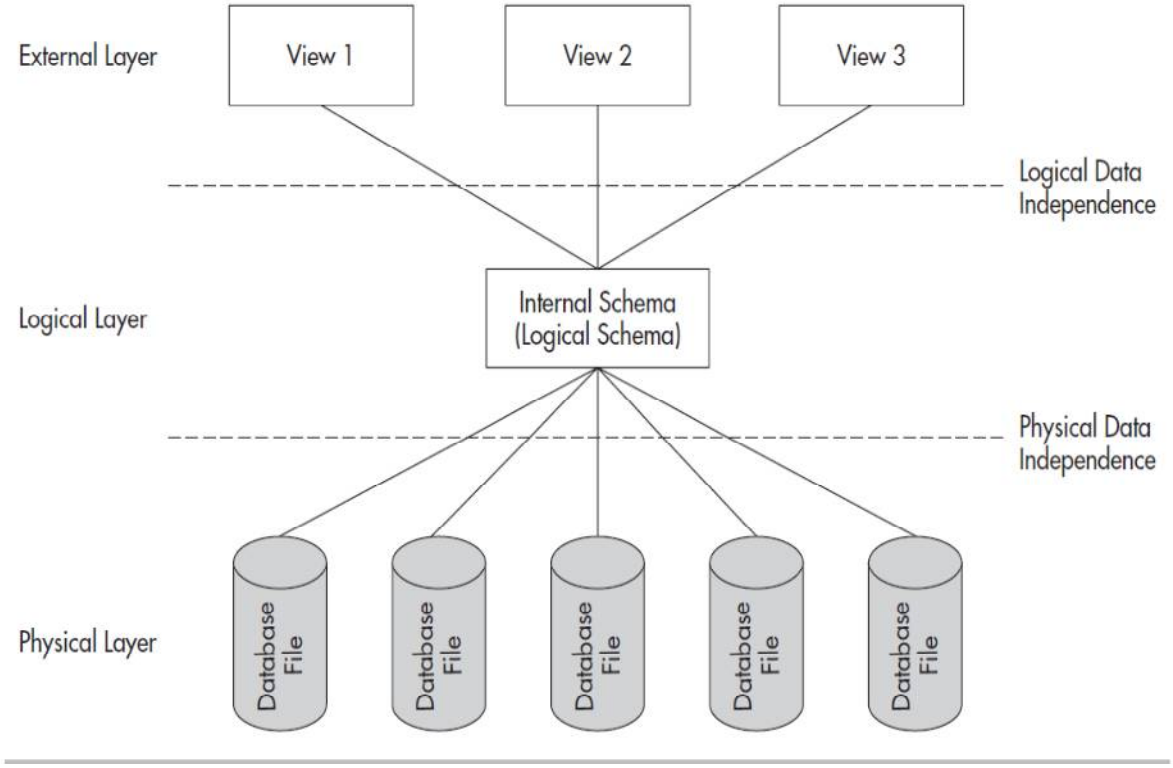


Figure 2. Database Layers of abstraction (Oppel 2009, 7)

The physical layer controls all data files that hold all the data for the database. The Logical layer consists of the first two abstraction layers in the database. The physical layer has a solid subsistence in the operating system files while the logical layer is an abstract data structure put together from the physical layer as required. Additionally,

external layer is poised of the user views and it is the second layer of abstraction. (Oppel 2009, 8-9.)

3.3 Introduction to Database Development

When developing a database, the framework for which database are designed as to be understood (Oppel 2009, 172). These frameworks are databases development life cycle (hereinafter DDLC) or software development life cycle (hereinafter SDLC).

Oppel (2009, 172) maintains a DDLC consist of all the occurrence that arise from the moment one identifies there is a need for a database, through is development and operation, and it ends when the database is no more in use. However, Murthy (2008, 46) refers to the “database development life cycle as a process for designing, implementing and maintaining a database system”. Murthy (2008, 59) perceives SDLC as the segmented approach to analysis and design in which systems are best developed following a precise cycle or analysis or user action. In other words, SDLC is series of phases whereby unique kind of work is done.

Oppel (2009, 174) identifies the phases as illustrated in the diagram below as the traditional system development life cycle for database development.

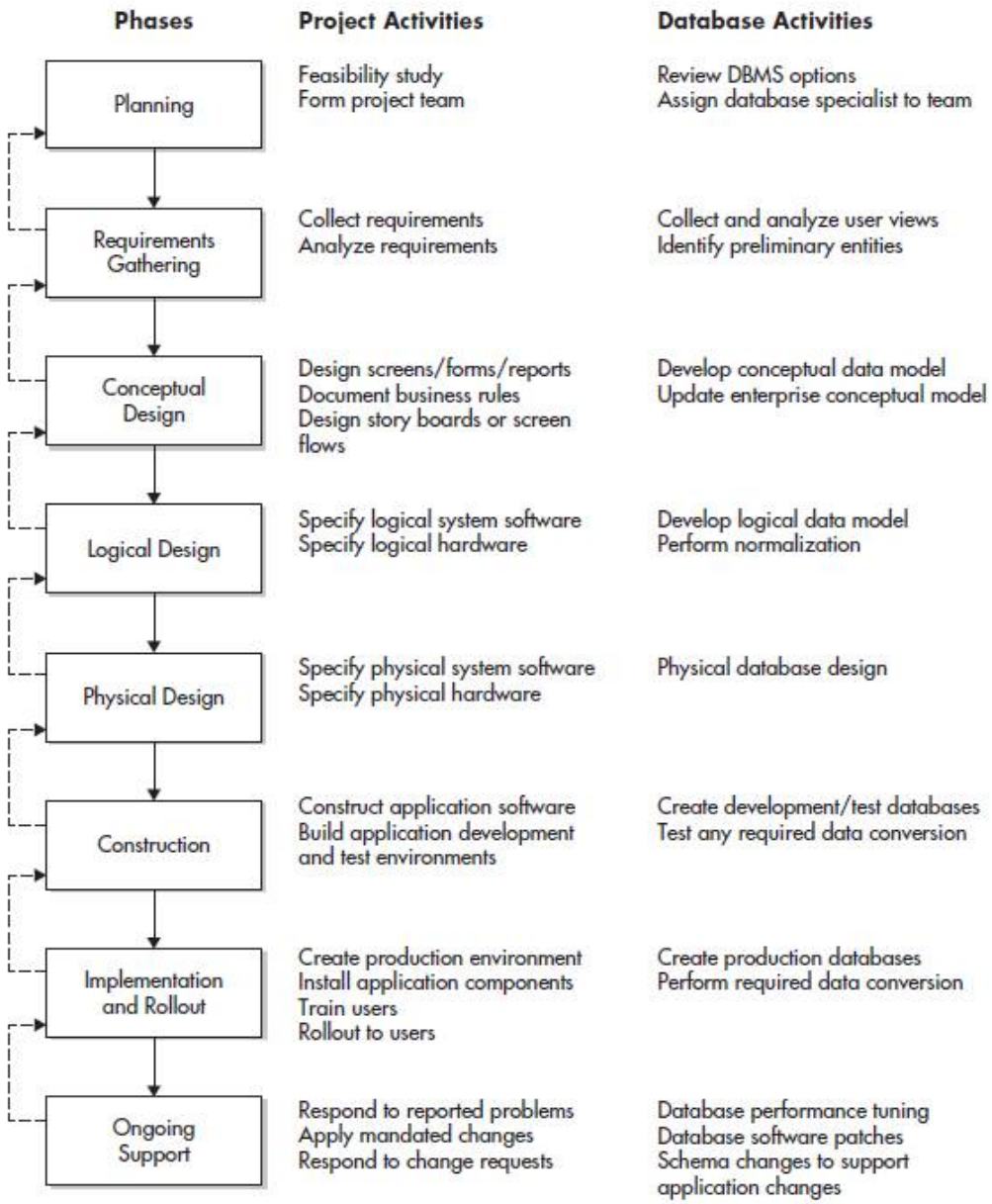


Figure 3. Traditional system development life cycle (Oppel 2009, 173)

I adopted the SDLC framework for the design of database for the game. Meyer (1998 as cited by Murthy 2008, 51) argues that to enlarge the effectiveness, system developers have to strictly look at the theory underlying the traditional SDLC. SDLC methods according to Meyer (1998 as cited by Murthy 2008, 51) support function-oriented approach. This approach views the system from the perception of its functions rather than of the data. Figure 3 as illustrated in the diagram above and discussed on the

database design perspective below is a traditional SDLC and it was adopted from Oppel (2009, 173.)

At the planning phase, the database activities entail the analysis of the DBMS options and establishing if the present technologies in use meet the general need for the project. In this phase of the database design, I was not opportune to critically analyze of the previous DBMS of the previous game. The case company gave me some requirements and specification for the game, and after the review of the requirements and specification, I chose SQLite because of its properties. My previous experience with MySQL also influenced my choice for it.

At the requirement gathering phase, the item most required in this phase is the user views. The user view is the technique fashioned to the needs of that person or application and this can be conducted by interviews, survey, observation, and document survey. At the inception of the project, I conducted an interview with the supervisor on the recommendations of the case company on the user views of the formal game and the specification needed from the case company perspective to improve the game.

At the conceptual design phase the database specialist or database administrator designated to the project updates the enterprise conceptual data model. The model is typically maintained in the structure of an entity-relationship diagram (hereinafter ERD). At this phase of the database design of the game and with the case company requirements and specification, entities were added for better functionality of the game. A rough sketch of the ERD is shown in Figure 4 below.

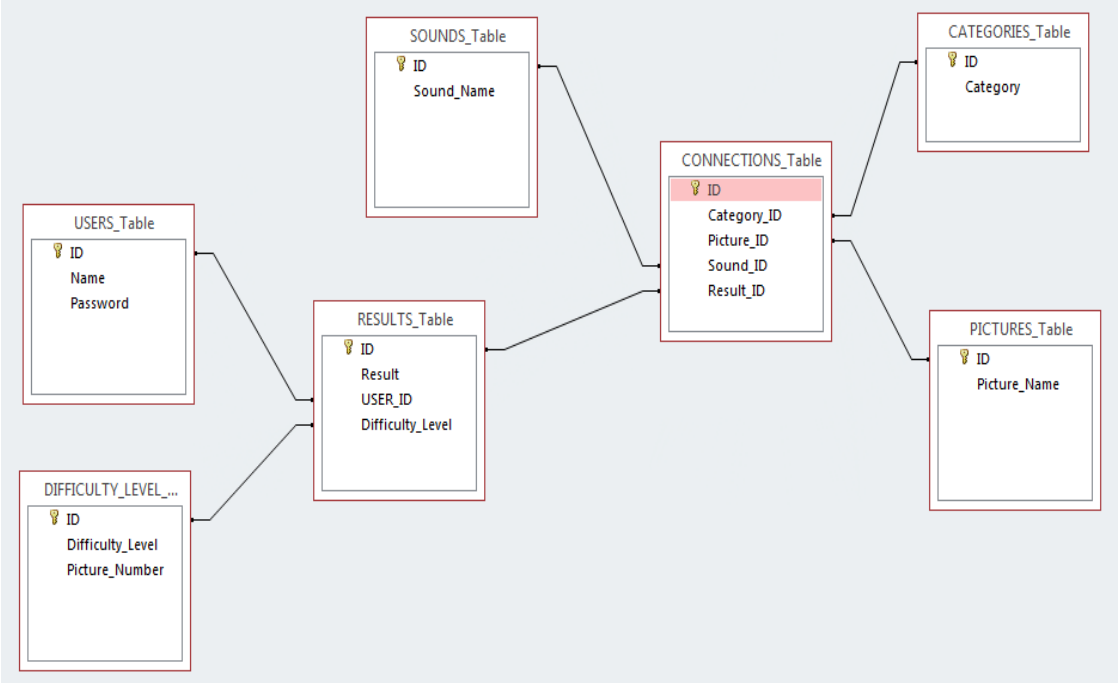


Figure 3. A sketch of ERD

At the logical design phase, the main aim is normalization, a technique developed by Codd (1972 as cited by Oppel 2009) for designing relational database tables that are the most suitable for transaction-based systems. Oppel (2009, 190) maintains that "Normalization is a technique for producing a set of relations (data represented logically in a two-dimensional format using rows and columns) that possesses a certain set of properties". Figure 5 as adopted from Oppel (2009, 191) shows the normalization process.

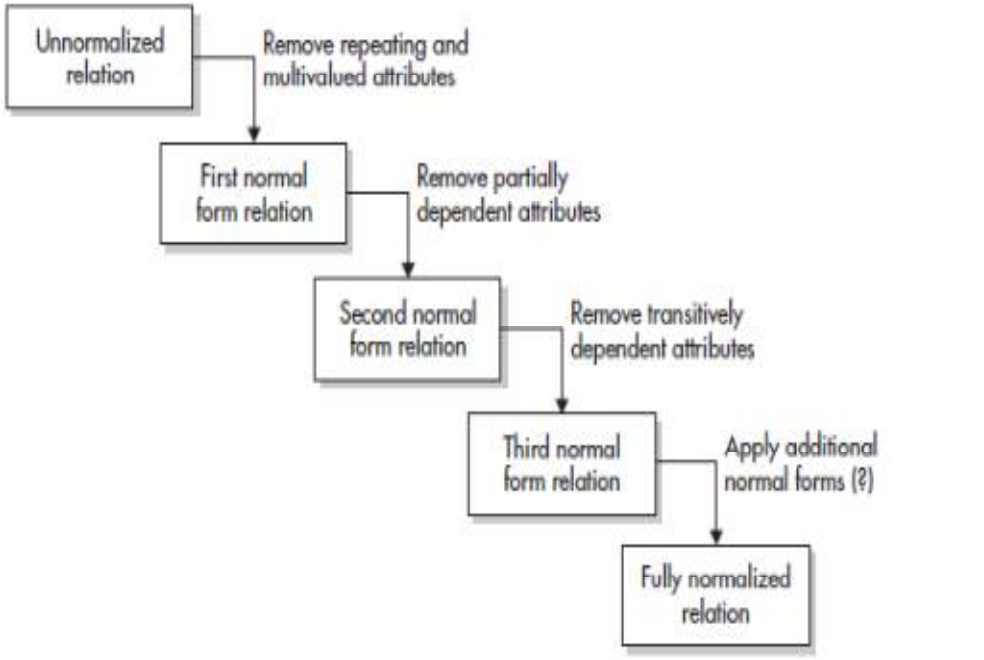


Figure 5. The normalization process (Oppel 2008, 191)

At the physical design phase, the normalized relations that were designed before the logical design phases are put to use in the relational DBMS(s) to be used. Specifically, Data Definition Language (hereinafter DDL) is created to describe the database objects, in addition to the SQL clause that describes the physical storage of the tables and indexes. At this phase, there is no physical design of the database because the game has not been implemented.

At the construction phase, the whole database system is pulled together and tested from one end to another. The database cannot be tested at this phase because there is no physical implementation for the database at the moment.

The implementation and the roll out of the game will be discussed with the company at later a day after the three major parts of thesis work are completed and designed and the implementation is actualized.

Ongoing Support is assigned to the support team when a database has been implemented. The support team is the team that will respond to any issues that may arise from the implementation of the application. The game has not been implemented and therefore, there is no ongoing support. However, it will be discussed with the case company after the implementation of the game.

3.4 Understanding Relational Databases

“Relational model represents a database as a collection of relations. It is a way of looking at data that is, it is a prescription for a way of representing data (namely be means of tables), and a prescription for a way of manipulating such a representation (namely, by means of operators such as JOIN). Thus, it is a data model representing data in the form of tables, consisting of rows and columns of data” (Murthy 2008, 95). The relational model has three aspects of data i.e data structure, data integrity, and data manipulation.

In a relational model, there is some common terminology used, a row is termed tuple, a column header is termed attribute and a table is termed relation. The data type describes the types of values in each column is called domain.(Murthy 2008, 95.)

3.5 Choosing Relational Database System

Several database models have emerged and many have gone extinct as the database revolutionized. An example is the Hierarchal model and Network model. The foremost ones in the market are relational, object-relational and the new paradigm of databases called Not Only SQL (hereinafter NoSQL). NoSQL and SQL databases are of accompaniment to each other rather than rivals.

Relational database management systems are one of the frequently used DBMSs in the world. Most of the institution, organization and personal computers use a part of the software that relies on RDBMSs. These software applications utilize relational database

through an assigned databases server or a lightweight RDBMS engines that are embedded in the software applications. (Juba et al 2015, 3-5.)

There are couples of relational database system but only a few supports the Android platform. These are SQLite, PostgreSQL, and MySQL e.t.c. PostgreSQL is an open source object-relational database management system. It accentuates on extensibility, creativity, and compatibility. It competes with other relational database vendors and can be used by different sectors like government establishments, public and private sectors. (Juba et al 2015, 31.)

MySQL is an SQL client/ server open source RDBMS that originated from the Scandinavia, and It was well accepted because of is swiftness and easy to use properties. Suehring & Valade (2013, 449) refers to MySQL as the most used database for websites that are written PHP. And further states MySQL and PHP work together because MySQL serves as the manager in the database system. MySQL might be one of the most used SQL RDBMS for a website but SQLite still holds an upper edge due to the bigger proportion of it been used for Smartphone and embedded software.

The choice of SQLite from the numerous relational database systems was based on the analysis done on the requirements from Datero Oy and the resource constraints for the game database design. SQLite is an open source DBMS, was also selected because of is versatile and his easy to use properties.

SQL is a language designed for managing data in relational databases. SQLite is a scaled down version of MySQL, PostreSQL, and other known database systems. Owen (2006, 1) defined SQLite as an open source embedded relational database. However, Kreibich (2010, 1) refers to SQLite as a public-domain software package that offers a relational database management system (herein RDBMS) and he further explains that “Lite” in SQLite does not mean its ability or potentials but rather it is lightweight when using it for setup complexity, administrative overhead, and resource usage.

The choice of SQLite for the database setup of the game was because of its characteristics or properties to other open source RDBMS such as MySQL and PostreSQL. Kreibich (2010, 1) explains these properties as is discussed below.

The server-less property means that SQLite does not need a separate server procedure to function because It accesses its storage files directly because the entire database engine is integrated into the application that needs to access the database. By eliminating the server from the application, a considerable quantity of complexity is removed, which makes software elements basic and practically removes the need for advanced operating support.

The difference between a traditional RDBMS and SQLite is illustrated in figure 6 and 7. Figures 6 was adopted from Kreibich (2010, 3) display an architecture traditional RDBMS server.

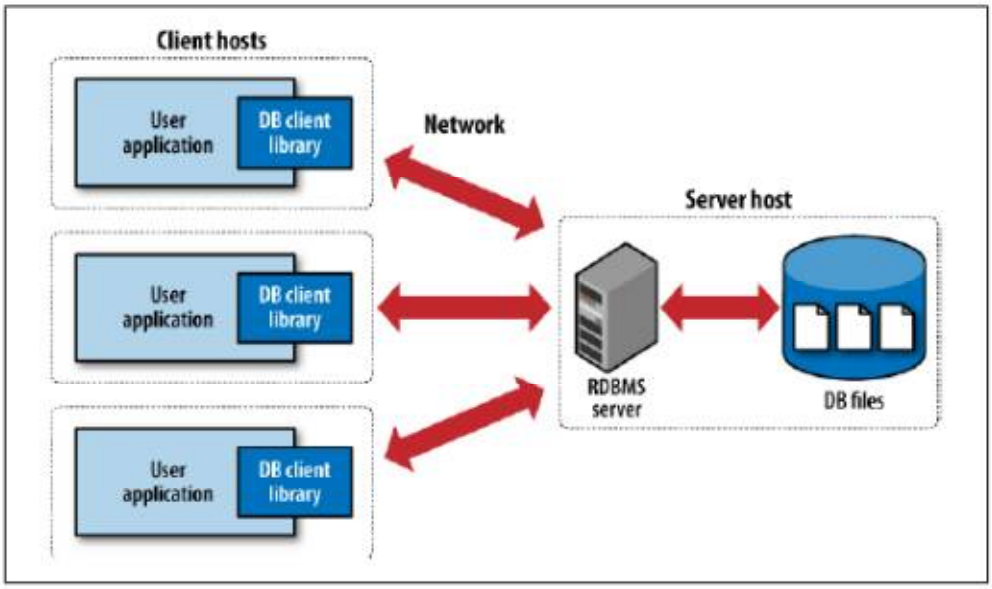


Figure 4. Architecture of traditional RDBMS Kreibich (2010, 3)

Figure 7 below was adopted from Kreibich (2010, 3) display an architecture of SQLite server-less databases system.

The simplicity property makes it possible to directly port SQLite in any environment and mobile phones, game consoles, handheld media players and many other devices, where other database systems cannot do.

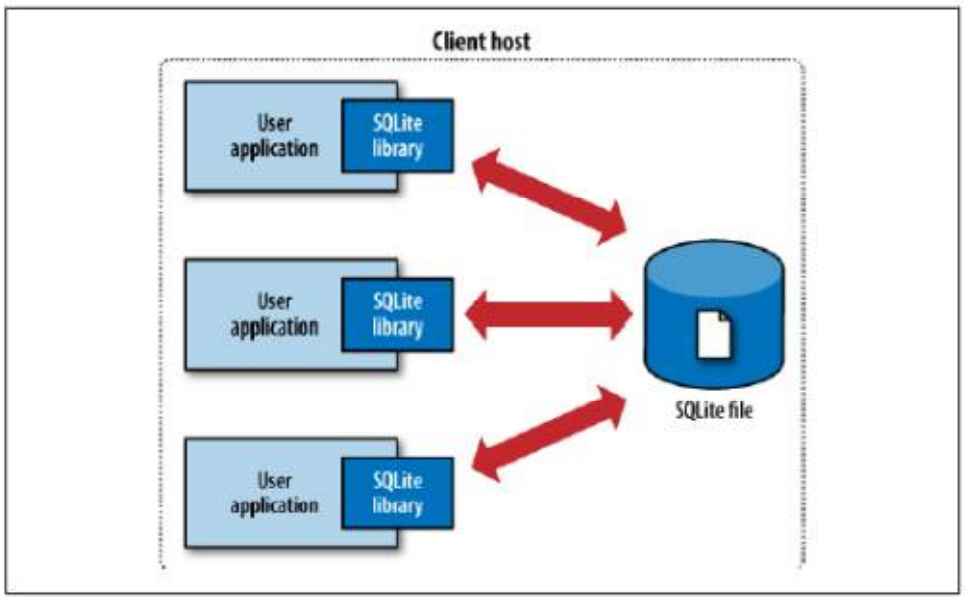


Figure 5. SQLite server-less architecture Kreibich (2010, 3)

In Zero Configuration, there is no need for any configuration since SQLite does not make use of a server, and therefore, the end user would not know they are using any database because the database engine is directly integrated into the application.

The cross Platform property means, the whole database is in a single cross-platform file that needs no administration. The cross platform is becoming an increasingly used approach in delivering the application to the different mobile platform.

The self-contained property of SQLite means a single library is controlling the whole database system that is directly incorporated into the host application of the memory game. The SQLite, unlike the majority RDBMS, does not contain a client/server architecture.

The Small Runtime footprint property means building the default needs less than a megabyte of code and a few megabytes of the storage or memory. The library and memory can be considerably reduced with little adjustment.

The Single file database means the whole database is in a single file and it contains the database layout with the main data input in different tables and indexes. Its file format is on a cross-platform, and therefore it can be used on any machine.

4 MEMORY GAME DATABASE DESIGN

4.1 Introduction to the Memory Game Database Design

The database design of the memory game will be embed into a mobile application to make the database fully functional. The new feature will be added to the new game to make it more attractive to the end user. The sound feature added to the game enhances the experience of users with sight challenges.

4.2 Understanding Tables and Attributes

Davidson and Moss (2012, 10) refers to a table as an object that will be engaged in your design and code. And further, explain a table as been used to store information and it signifies something that you want to store data about. A table can represent people, things, places or ideas about the information that need to be stored.

Davidson and Moss (2012, 14) refers to attribute as a familiar term in the programming world. It states information about an object and in the logical modeling, it can be related to anything that signifies entities.

The tables and attributes in the memory game database design are illustrated and described below.

Users table, this stores the player information to the game. This information is the attributes on the table, which are the user Id, name, and password, these attributes are data types saved for each player of the game to keep the record of those who have played the game. The table 2 below is an illustration of the users table.

Table 2. Users table

```
CREATE TABLE `USERS_Table` (
        `user_ID`       INTEGER,
        `Name`          TEXT,
        `Password`      TEXT,
        PRIMARY KEY(user_ID)
);
```

Categories table these are the different types of categories in the memory game. The attributes on the table are the category id and category name. The table 3 below is an illustration of the category table.

Table 3. Category Table

```
CREATE TABLE `CATEGORIES_Table` (
        `Category_ID` INTEGER,
        `Category_Name`         TEXT,
        PRIMARY KEY(Category_ID)
);
```

Connections table connects all the tables together. The attributes in this table are the connection id, category id, picture id, sound id and result id. The table 4 below is an illustration of the connection table.

Table 4. Connection table

```
CREATE TABLE `CONNECTIONS_Table` (
        `Connection_ID`         INTEGER,
        `Category_ID` INTEGER,
        `Picture_ID`    INTEGER,
        `Sound_ID`      INTEGER,
        `Results_ID`    INTEGER
);
```

Difficulty level table, at this table the player choose the amount of challenge they will encounter while playing the game. The attributes at this level are the difficulty id, difficulty level, and picture number, these attribute data types are also saved along with result because it shows the level each player have reached along with the result achieved. The table 5 below is an illustration of the difficulty level.

Table 5. Difficulty level

```
CREATE TABLE `DIFFICULTY_LEVEL_Table` (
                `Level_ID`      INTEGER,
                `Difficulty`    TEXT,
                `Picture_Number`                INTEGER
);
```

Pictures table, this is the table that shows the different pictures that will be paired in the game. The attributes in this table are the picture id and picture name. The table 6 below is an illustration of the pictures table.

Table 6. Pictures table

```
CREATE TABLE `PICTURES_Table` (
                `Picture_ID`    INTEGER,
                `Picture_Name` TEXT,
                PRIMARY KEY(Picture_ID)
);
```

The result table shows the result of each game played. The attributes in this table are the result id, result, users id and difficulty level. The data types are this table are saved because it keeps the record of the outcome or result of each game played. The table 7 below is an illustration of the results table.

Table 7. Results table

```
CREATE TABLE `RESULTS_Table` (
                `Results_ID`    INTEGER,
                `Results`       INTEGER,
                `USERS_ID`      INTEGER,
                `Difficulty_level`                  INTEGER,
                PRIMARY KEY(Results_ID)
);
```

Sound table, this table shows the different sounds on the game. The attributes on this table are sound id and sound name. The table 8 below is an illustration of the sound table.

Table 8. Sound table

```
CREATE TABLE `SOUNDS_Table` (
                `Sound_ID`      INTEGER,
                `Sound_Name`    TEXT,
                PRIMARY KEY(Sound_ID)
);
```

4.3 Memory Game Database Design

Stephens (2009, 3) states that a database is not useful if it can retrieve data quickly, reliably and consistently. It will not be meaningful if it is full of incorrect or contradictory data.

This impending problem can be solved by using modern tools, a good database design and applying common sense and having some knowledge of what the problems are. (Stephens 2009, 3). The figure 8 below is an illustration of the database design of the memory game.

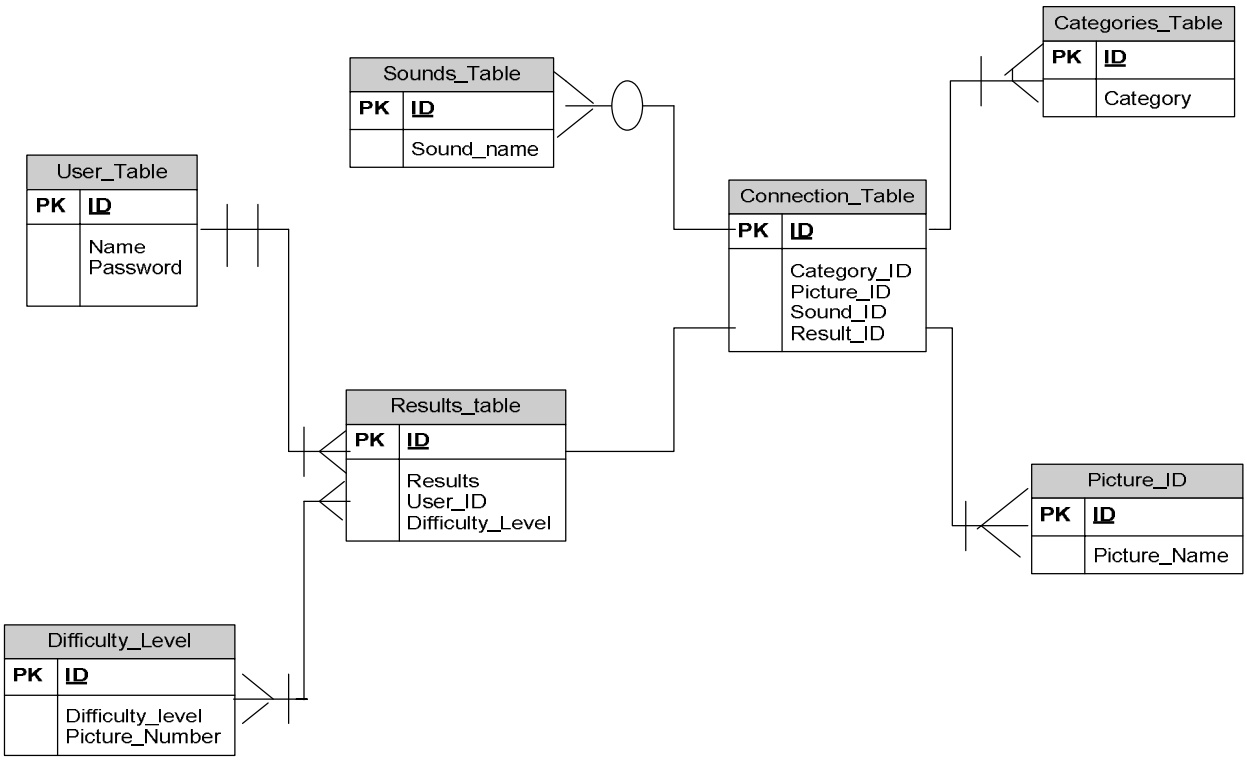


Figure 8. Memory game database design

5 ANDROID

5.1 Introduction of the Android

The Android was introduced in 2005 after Google bought a small establishment called Android Inc. For this reason, there has been assumption Google was interested in the mobile device space. In 2008, after the publication of version 1.0 of Android, it put an end to the assumption. It was time for it to compete with the other platform that has been established before the introduction of Android. An example of these platforms is iPhone, Blackberry, and Window operating systems. Android popularization in the market has been exceptional. (Zechner & Green 2012, 2.)

5.2 Open Source Platform

The Android is an open source platform. The whole stack, starting from low-level Linux modules to the local libraries, and developing the whole applications, is completely free.

The Android was certified under business-friendly licenses (Apache/MIT), it can freely be expanded and be use for various purposes. Other third-party open source libraries brought into the Android stack were rewritten under new Agreement terms.

The Android can be used or modified at any time because there is no need to license it without any string attached to it. At various levels of the platform, it has many hooks attached to it, this allows anyone to extend it in unforeseen ways. (Gargenta 2011, 2.)

5.3 Android Versions

Since 2008, when Android was made public, it has on regular bases received updates, with most its names, named after deserts with the exemption of Android 1.1 that is immaterial as of today. A good number of Android platforms have new functions, that

are structured like application programming interface (APIs) or latest development tools that are relevant for game developers.

Version 1.5 (Cupcake) support added designed for local libraries in Android applications that were earlier constrained to being in written in Java. A local code can be favorable in conditions where performance is the greatest concern.

Version 1.6 (Donut) pioneer supports for diverse screen resolutions. This resolution supports lower and higher resolutions screens. The present phones have wide video graphics array (hereinafter WVGA) screens (800 * 480, 848 *480, or 852 * 480 pixels) and a few low-end devices support quarter video graphics array (hereinafter QVGA) screens (320 * 280 pixels).

Version 2.0 (Éclair) is an additional support for multi-touch screens. It also has some new features such as expanded account sync and this feature makes it possible for users to have several accounts on a device for synchronization of contact and emails.

Version 2.2 (Froyo) Added just-in-time (JIT) combined to Dalvik virtual machine (VM), the software that controls all java applications on Android. JT increases the pace of Android application execution in consideration of the situation.

Version 2.3 (Gingerbread) added the latest coexisting garbage collector (hereinafter GC) to the Dalvik VM. It has the propensity to propel the developers a little passionate most time. The GC can easily coexist in daily game development.

Version 3.0 (Honeycomb) was made public in the year 2011 and it produced a tablet version of Android, Honeycomb had more considerable API modifications than any other Android version released till date. Honeycomb 3.1 versions had additional broad support for dividing and controlling high-resolution tablet screen. It added features like USB host support and USB peripherals support, keyboards, and joysticks that are PC-like features. Its major challenge was it is targeted at tablets.

Android 4.0 (Ice Cream Sandwich (ICS)) is the merger of Honeycomb (3.1) and Gingerbread (2.3) with a similar set of features that works on both tablets and phones. ICS has a new APIs, and as at 2011 eminent Android device manufacturer s concur to support the newest android operating system for a gadget for the period of eighteen months for the purpose of fragmentation and it also means the new APIs on ICS will be accessible on most phones faster . Fragmentation means different things to both the end users and developers. To the end user, it signifies not being able to use some applications and features due to the old versions of android cannot upgrade to the newest version and to the developers it signifies some concern as to be considered when developing an application that will work on all android versions.

The new APIs on ICS will be an advantage to put into consideration when developing the new memory game after the whole parts of the game design has been completed. This was discussed with the other members of the team and we all decided it will be and hedge for the memory game in a saturated mobile market.

Android 4.1 (Jelly Bean) the way UI is developed is improved, and all general the interpretation. The exertion is known as “Project Butter”. Nexus 7 tablet was the first device to feature Jelly Bean. (Zehnner & Green 2012, 2-7.)

Table 9. Nicknames of different versions

Android version	API level	Nickname
Android 1.0	1	
Android 1.1	2	
Android 1.5	3	Cupcake
Android 1.6	4	Donut
Android 2.0	5	Eclair
Android 2.01	6	Eclair
Android 2.1	7	Eclair
Android 2.2	8	Froyo (frozen yogurt)
Android 2.3	9	Gingerbread
Android 2.3.3	10	Gingerbread
Android 3.0	11	Honeycomb

Table 9. Nicknames of different versions

The table 2 above as adopted from Gargenta (2011, 4) is a tabular illustration between different versions.

Android improvements are shown in is version numbers but the most important is not the version numbers but the API level. The versions change most time because the APIs levels as changed or because of minor bug fixes or performance improvements.

An application developer will always need to know the API level is application is focusing on in order for it to run. The main function of an API is to determine the type of device that runs your device on not.

As application developers will always want their applications to run on as many devices as possible. Having that in mind, they try to focus on API level that is low, in consideration of the distribution of Android versions on devices. The Figure 9 shown

below as adopted in Gargenta (2011) is a snapshot of the Android Device Dashboard from mid-2010.

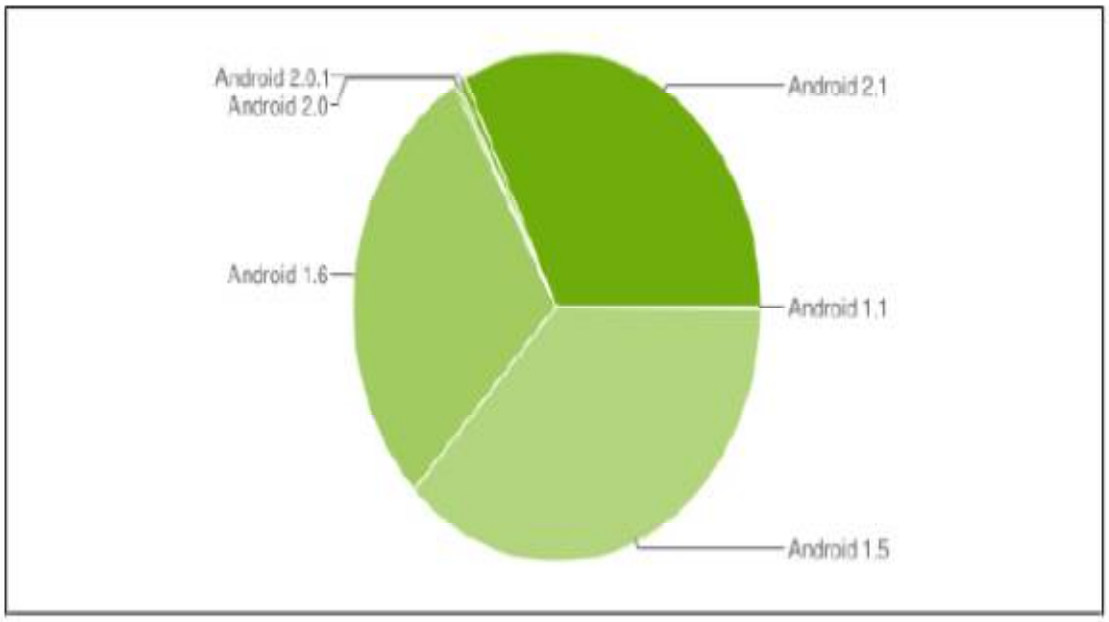


Figure 9. Historical Android versions distribution through January 2011 Gargenta (2011, 5)

6 SQLITE AND SQL

6.1 Introduction of SQLite

SQLite was envisaged on a battleship. The author of SQlite, D. Richard Hipp, was at then working for General Dynamics on a program for the U.S Navy developing software to be used on board guided missile destroyers.

It was launched in 2000, as an open source embedded relational database. The aim of its design was to proffer a suitable approach for applications to control data without operating cost that usually come with exclusive relational database management system. (Owen 2006, 1-3.)

6.2 Mobile and an Embedded Database

SQLite concurrently runs in the application it serves within the allotted space, rather than running separately like other DBMS. That is why it is called an embedded database. To an outsider, it won't be obvious the program, had an RDBMS running on the device. (Owen, 2006,1.)

The power and capability of smartphones, mobile devices, and other embedded systems keep on growing the devices are able to deal with bigger and more complex data. Several mobile devices concentrate around searching, categorizing and displaying a considerable amount of structured data. It might be something as fundamental as an address book, and something as difficult as mapping and route applications.

SQlite provides a relational model that fits application requirement and data storage. SQLite is a comparatively small and resourceful product that runs well in a controlled environment. It a database-in-a-file model also makes it simple to copy or support data stores effortless and quickly. It not a surprise most major smart phones software

development kit (hereinafter SDK) supports SQLite and allows it to be effortlessly compiled for their platform. (kreibich 2010, 176.)

6.3 Memory

Mobile devices have limited memory resources. At the later days, after the whole design phase of the memory game as be completed. The team and the commissioning company have decided to develop the game, the team will need to take in cognizance the memory usage, and frequently need to be conscious of the resources that might be consumed. The greater part of SQLite memory usage comes from page cache and the memory can be controlled by selecting a page size and cache size, Open or attached database ordinarily has its own independent cache (kreibich 2010, 176).

6.4 SQL Functions and Extensions

SQLite permits a developer to expand the SQL environment by creating custom SQL functions. These functions are utilized in SQL statements and the code to execute the functions is written in C.

There are three types of custom functions that are supported by SQLite. The first supported custom function is the simple scalar, it takes a few set of parameters and return a single value. An example would be built in function abs(), it receive a single numeric parameter and returns the complete value of the number.

The second type of function is an aggregator. They are SQL functions, such as sum(), or avg(), are applied in conjunction with group by clauses to get the cumulative sequence of values together into a final result.

The final type of custom function is collation. They are used to define custom variety for an index. Theoretically, collations are simple, and they receive two text values and return a greater than, less than or equal status. (Kreibich 2010, 181.)

6.4 Users of SQLite

As at today, SQLite can be used by with different software and products. It is been used by Apple's OS X operating system as one of their Core Data application frameworks. The other system's that utilize SQLite are Safari web browser, Mail.app email program, RSS manager and Apple's Aperture photography software.

SQLite exclusively can be found in a database supporting the Service Management Facility that is introduced with Solaris 10, in Sun's Solaris operating environment, a primary component for predictive self-healing technology.(Owen 2006, 4.)

7. CONCLUSION

A database can be a controlling tool with the responsibility of doing what exactly the computer programs do most. Which are to store, manipulate and display the data. Except a database stores its data efficiently, the whole application will be ineffective not minding the level of the design of the other part of the application.

These impending problems can be addressed by using a recent database tool a good database design represent one of the major parts of achieving a good application.

In the present day global market for mobile game application, the challenge of choosing the right database management system has been one of the biggest challenges for database developers. However, considering the different database languages, the database developer has to have these attributes.

a) A clear understanding of what the databases will serve.
b) The compatibility of the databases with the purpose it will serve.
c) The technical know-how of database developer.

All the listed attributes will be a guiding principle of selecting the right choice of database management system. Having an understanding of all these attributes will also give an answer to the first question of this research work, “How does database design framework affect the accessibility of data on a mobile application for the Android platform?”. With that understanding, there is a high possibility of a good design framework which will guarantee a good database design.

Even though the research work did not discuss in full details the other databases design framework, the research study suggested that a databases framework serves as a foundation for any database design, and without a well-organized foundation, there is a high possibility that the database design will be a failure. With a greater part of all application relying on databases, it is expected that every person involved in application

development should have a concrete knowledge in the database design and construction.

At the inception of the research work, an interview was conducted with one of the supervisors by the Juha Meriläinen as a means of data gathering. From the interview conducted, the research work was able to know the stakeholder view about the game and their stake in the previous game that could not be implemented due to the hitches. One of the hitches of the previous was the data were not functioning at is best.

According to the findings of the research work SDLC framework support a function-oriented database design and the phases of SDLC framework phases for the database design was explained. SLDC framework was suggested to the assignor, for the framework of database design and SQLite DBMS which is compatible with the Android platform.

The objective of this thesis work was designing a database with SQLite language for the Android platform Memory Game. The thesis study also highlighted the advantages of developing a database with SQLite language for the Android platform on a mobile application. The thesis also proffers solution to the database hitches of the previous game by using SDLC framework and SQLite DBMS.

BIBLIOGRAPHY

Davidson. L & Moss. J. M 2012. Pro SQL Server 2012 Relational Database Design and Implementation. United States: Apress. Accessed 3 April 2016 http://it-ebooks.info/book/1056/

Dodig-Crnkovic, G. 2010. Constructive Research and Info-Computational knowledge Generation. School of Innovation, Design and Engineering, Computer Science Laboratory, Malarden University, Sweden. Accessed 12 December 2015. http://www.mrtc.mdh.se/~gdc/work/MBR09ConstructiveResearch.pdf

Gargenta, M. 2011. Learning Android. United States: O'Reilly Media, Inc. Accessed 20 November 2015. http://it-ebooks.info/book/621/

Jha, N. K. 2008. Research Methodology. New Delhi: Abhishek publication. Accessed 20 January 2016. http://ez.lapinamk.fi:2054/lib/ramklibrary/reader.action?docID=10416498

Juba . S. & Vannahme, A & Volkov, A. 2015. Learning PostreSQL. Birmingham United Kingdom: Packt Publishing. Accessed 4 April 2016 http://it-ebooks.info/book/6571/

Kreibich, A. J. 2010. Using SQlite. United States: O'Reilly Media, Inc. Accessed 30 December 2014. it-ebooks.info/book/147/

Lukka, K. 2000. The Constructive Research approach. Metodix. Accessed 12 December 2015. https://metodix.wordpress.com/2014/05/19/lukka-konstruktiivinen-tutkimusote/

Murthy C. S. V. 2008. Database Management Design. Mumbai: Himalaya Publishing House. Accessed 2 January 2015. http://ez.lapinamk.fi:2054/lib/ramklibrary/detail.action?adv.x=1&docID= 10415516&f00=all&f01=subject&p00=database+development +process&p01=%22Database+Management%22

Oppel, A. J. 2009. Database: A Beginners Guide. United States: McGraw-Hill Companies. Accessed 8 January 2015. it-ebooks.info/book/1911/

Owens, M. 2006. Definite Guide to SQlite. United States: Apress. Accessed 30 December 2014. http://it-ebooks.info/book/5088/

Powell, G. 2006. Beginning Database Design. United States: Wiley Publishing, Inc. Accessed 2 January 2015. it-ebooks.info/book/1161/

Sachdeva, J. K. 2009. Business Research Methodology. Mumbai: Himalaya Publishing House. Accessed 20 January 2016. http://ez.lapinamk.fi:2054/lib/ramklibrary/reader.action?docID=10416021

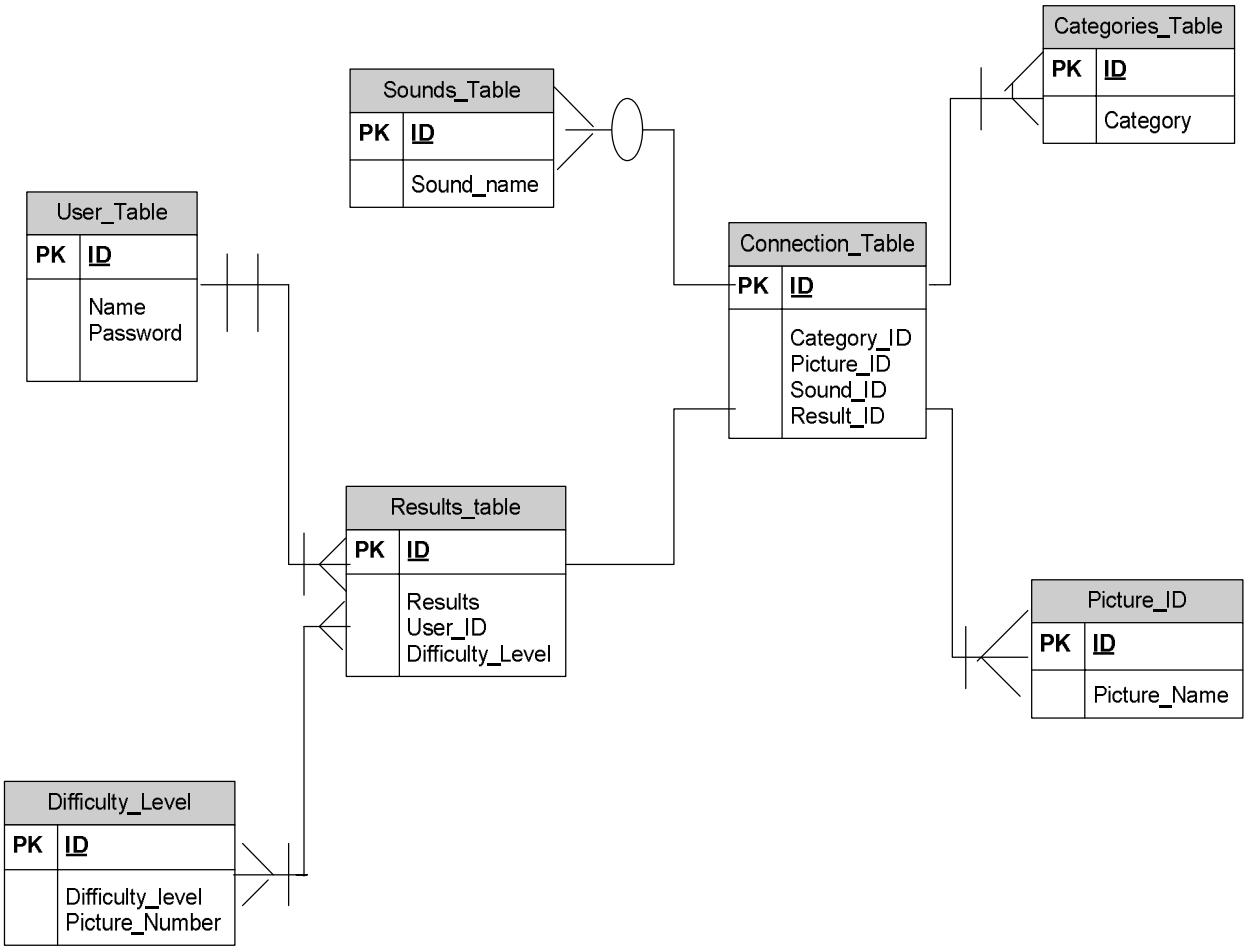
Stephens, R. 2009. Beginning Database Design Solutions. United States; Wiley Publishing Inc. Accessed 4 January 2015 http://it-ebooks.info/book/1464/

Suehring, S. 2013. PHP, MySQL, JavaScript & HTML5 All-in-One for Dummies. United States: Wiley Publishing Inc. Accessed 10 April 2016 http://it-ebooks.info/book/2083/

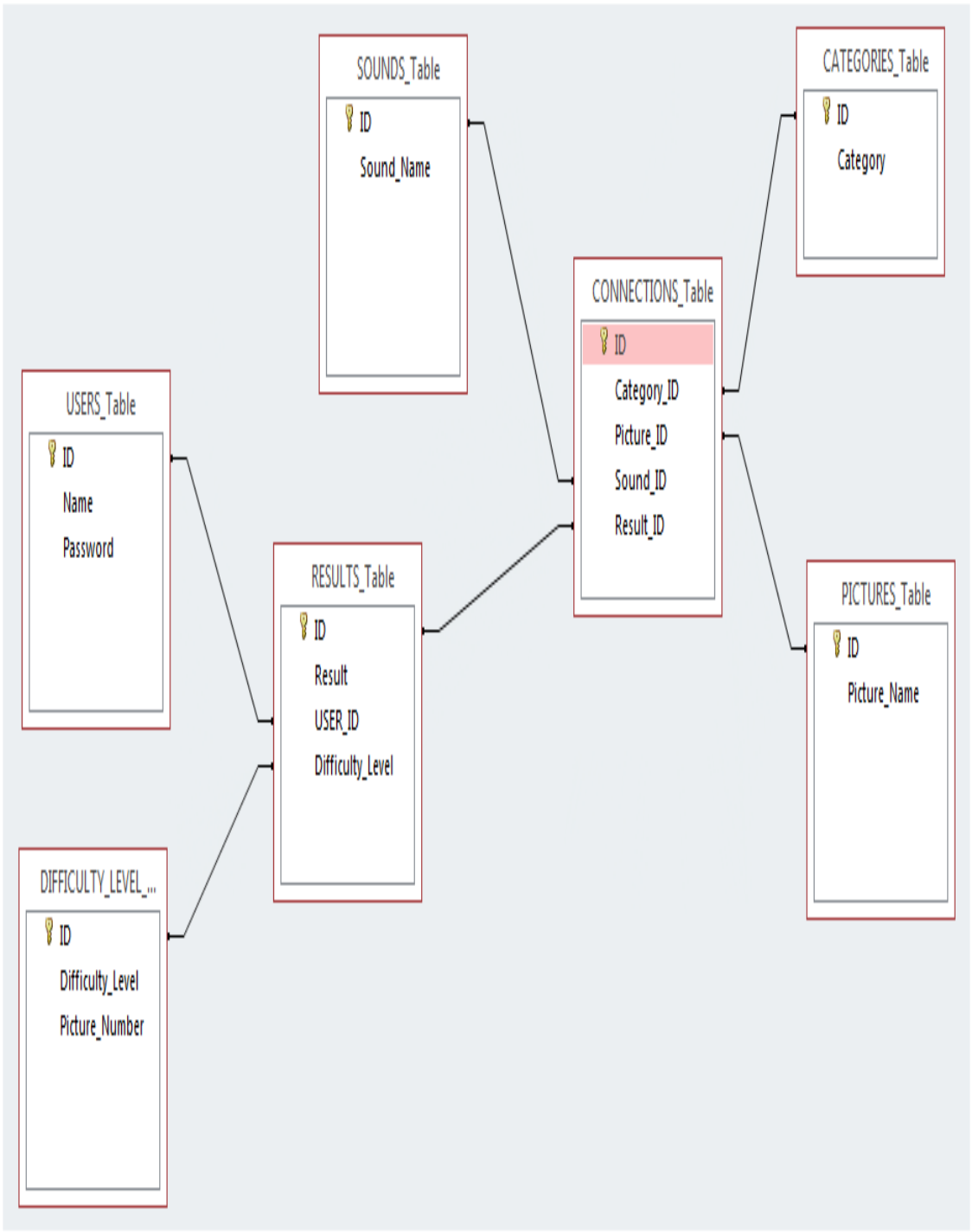
Zechner. M & Green. R 2012. Beginning Android games second edition. United States: Apress. Accessed 2 January 2015 https://it-ebooks.info/book/1305/

APPENDICES

Appendix 1. DATABASE DESIGN



Appendix 2. ENTITY RELATIONSHIP DESIGN



Appendix 3. DATABASE TABLE


```
CREATE TABLE `CATEGORIES_Table` (
                `Category_ID` INTEGER,
                `Category_Name`              TEXT,
                PRIMARY KEY(Category_ID)
);
```

```
CREATE TABLE `CONNECTIONS_Table` (
                `Connection_ID`              INTEGER,
                `Category_ID` INTEGER,
                `Picture_ID`  INTEGER,
                `Sound_ID`    INTEGER,
                `Results_ID`  INTEGER
);
```

```
CREATE TABLE `DIFFICULTY_LEVEL_Table` (
                `Level_ID`    INTEGER,
                `Difficulty`  TEXT,
                `Picture_Number`             INTEGER
);
```

```
CREATE TABLE `DIFFICULTY_LEVEL_Table` (
                `Level_ID`    INTEGER,
                `Difficulty`  TEXT,
                `Picture_Number`             INTEGER
);
```

```
CREATE TABLE `PICTURES_Table` (
                `Picture_ID`  INTEGER,
                `Picture_Name` TEXT,
                PRIMARY KEY(Picture_ID)
);
```

```
CREATE TABLE `RESULTS_Table` (
                `Results_ID`    INTEGER,
                `Results`       INTEGER,
                `USERS_ID`      INTEGER,
                `Difficulty_level`                  INTEGER,
                PRIMARY KEY(Results_ID)
);
```

```
CREATE TABLE `SOUNDS_Table` (
                `Sound_ID`      INTEGER,
                `Sound_Name`    TEXT,
                PRIMARY KEY(Sound_ID)
);
```