Assignment Cover Sheet - Group

Faculty of Science and Technology

**UNIT CODE: SIT302**

**ASSIGNMENT/PRAC No.: Assignment1**

**ASSIGNMENT/PRAC NAME:**

**Project Research and Investigation Report**

**DUE DATE: 5 December, 2016**

**Plagiarism and collusion**

Plagiarism occurs when a student passes off as the student’s own work, or copies without acknowledgment as to its authorship, the work of any other person.

Collusion occurs when a student obtains the agreement of another person for a fraudulent purpose with the intent of obtaining an advantage in submitting an assignment or other work

**Declaration**

We certify that the attached work is entirely the work of the group, except where work quoted or paraphrased is acknowledged in the text. We also certify that it has not been submitted for assessment in any other unit or course.

I agree that Deakin University may make and retain copies of this work for the purposes of marking and review, and may submit this work to an external plagiarism-detection service that may retain a copy for future plagiarism detection but will not release it or use it for any other purpose.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIRST NAME** | **FAMILY NAME** | **STUDENT ID** | **\*CONTRIBUTION**  **(%age)** | **DATE** |
| **XUN** | **SUN** | **215012871** | **100%** | **4 Dec** |
| **YUNHAO** | **ZHOU** | **215012174** | **100%** | **5 Dec** |
| **WANMIN** | **CHEN** | **215012989** | **100%** | **3 Dec** |
| **SAISANAS** | **PETER** | **213194708** | **100%** | **4 Dec** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

An assignment will not be accepted for assessment if the declaration appearing above has not been duly completed by the authors.

\* The contribution value shall be a percentage representing the amount of work completed by that group member in regards to their allocated task(s). For example, the contribution value for a particular group member shall be:

* 0% if that group member completed none of their allocated task(s),
* 50% if that group member completed half of their allocated task(s), and
* 100% if that group member completed all of their allocated task(s).

Project Proposal and Plans

SIT302 Project

Time Management and Activity Tracking

Client: Prof Paul Gastin

Supervisor: Dr. Mohamed Abdelrazek

Delivered on: Dec 5, 2016

Submitted by: SIT302 Team NO.16

Contact Email: zyunha@deakin.edu.au

Table of Contents

[1 Project overview 5](#_Toc468688210)

[2 Existing system overview 5](#_Toc468688211)

[2.1 Integrated planner research 5](#_Toc468688212)

[2.2 App development research 7](#_Toc468688213)

[2.2.1 User Interface 7](#_Toc468688214)

[2.2.2 Functionality 11](#_Toc468688215)

[2.3 Summary 13](#_Toc468688216)

[3 Methodologies Overviews 14](#_Toc468688217)

[3.1 Structured Systems Analysis and Design Methodology (SSADM) 14](#_Toc468688218)

[3.2 Scrum 15](#_Toc468688219)

[3.3 Object-Oriented Programming (OOP) 17](#_Toc468688220)

[3.4 Lean UX 18](#_Toc468688221)

[3.5 Methodology choice and justification 19](#_Toc468688222)

[4 Technologies and Resources Overview 20](#_Toc468688223)

[4.1 Planning phase 20](#_Toc468688224)

[4.2 Development phase 20](#_Toc468688225)

[4.3 Launch phase 21](#_Toc468688226)

[4.4 Testing phase 21](#_Toc468688227)

[References 22](#_Toc468688228)

[Appendix 23](#_Toc468688229)

# 1 Project overview

This project is aimed at developing a working prototype of a time management and activity tracking app. The quantification of the amount of time spent by individuals in various activities is of particular interest to researchers and practitioners. For athletes it might be about how much time is spent training and recovering, and how much load is being undertaken. For adolescent students, it might be about how much time is spent studying, in physical activity, in leisure activities, screen time and sleep.

This touch screen app would allow users to identify the types of activities to track and allow these to be ‘dragged and dropped’ into a 24 hour day or weekly calendar. Activities would be positioned and resized by touch. The back-end interface would need to database the data (and allow export) for subsequent analysis and visualisation over a selected period of time.

In this report, existing systems which have been researched and investigated will be described in this document. Besides, methodology has been chosen to guide project design and implementation, while the development, technologies and resources are also introduced in this document. It is benefit to complete the project research and investigation report because developers can access to existing systems which are similar with the project and it can give guidance for team members during the development of the project.

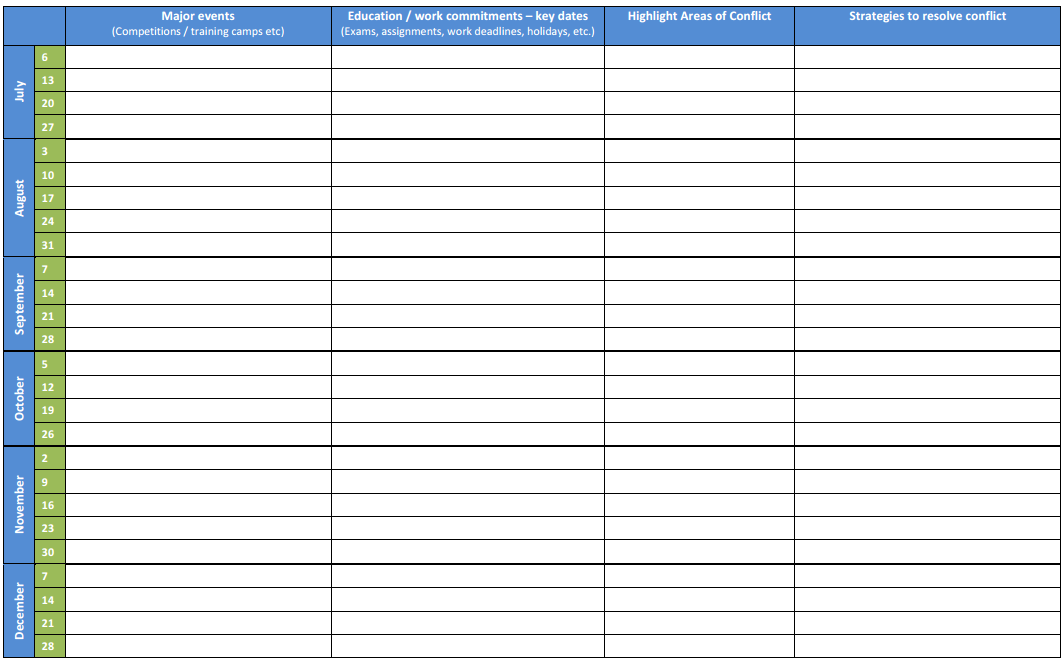
# 2 Existing system overview

Through analyzing the existing systems, both the app developer and User Interface designer can keep on track with the work they will do. There are two existing systems. One is an integrated planner which is produced by Australian Sports Commission (Manual), and the other is a mobile application – Hours.

## 2.1 Integrated planner research

Although integrated planner is manual, it still provides some functions which meets the demand of the client of this project. Our project is to make a mobile application to collect, record and quantify the amount of time spent by individuals in various activities. For these three functions, integrated planner needs people to collect the forms and record the amount of time spent by individuals.

Integrated planner provides a form for user to fill in. It uses the left-hand column to include all on and off field sporting commitments, and also shows potential sport commitments, the second column to show all education, work and lifestyle commitments, the third column to ‘red flag’ any conflicts and the forth column to make a note of the strategy used to resolve the conflict (Whipper, 2009).



***Table 1 Integrated planner***

It is important that the calendar view is provided in this table. However, our project is not a paper project, and it should be user-friendly. We decided to use a weekly timetable view to replace the calendar view. For ‘drag and drop’ function, we decided to use a vertical timeline to present a selected day of a week (to leave enough space for activities to drag).

## 2.2 App development research

For developing the mobile application of time management and activity tracking, all group members have analyzed the existing mobile application – Hours. It is an IOS application, featuring a visual timeline, smart reminders, and beautiful reports (which can be exported to PDF). Hours makes time tracking actually enjoyable. It includes a Today view widget and an intuitive timeline that helps user see what he/she has worked on and the time gaps you missed (Kennedy, 2015).

### 2.2.1 User Interface

1. ‘Today’ view

There are generally three main modules in the ‘Today’ view. They are header area, timeline area and activity area.

Header Area:

* The date of today
* An icon to open the calendar view
* An icon for settings

Timeline Area:

* Scales and makers
* Activities in different colors

Activity Area

* Blocks with the length of each activity
* Activity name and client
* Timer of each activity
* Add new timer

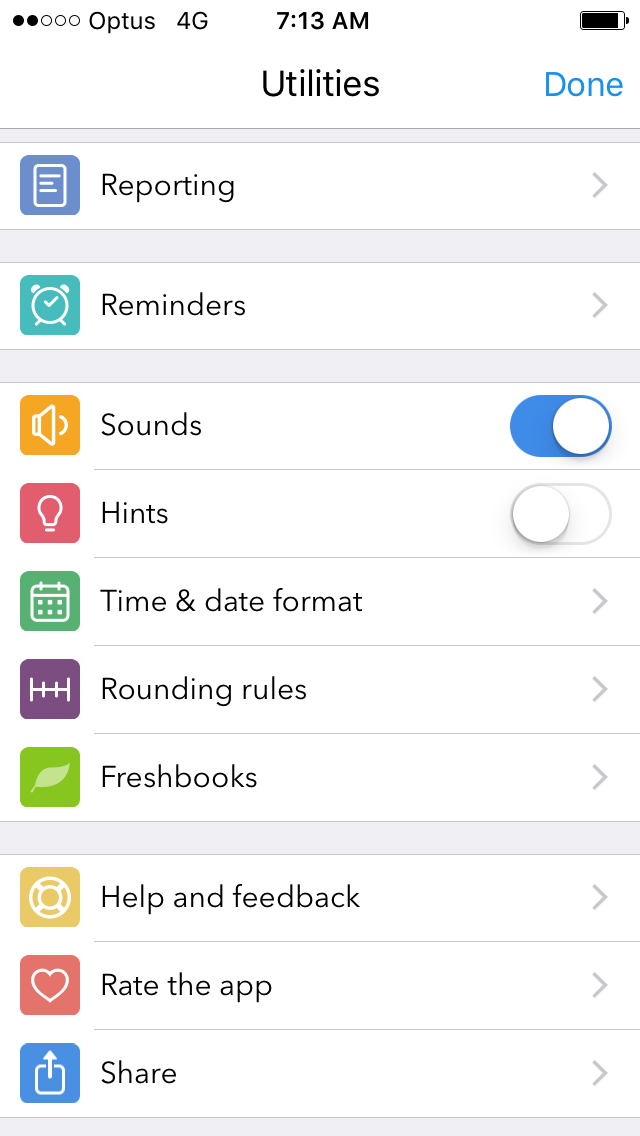
Here is the ‘Today’ view of Hours:



According to client’s requirements, the timer function is not necessary for our project. We should make a vertical timeline and an activity line on the other side.

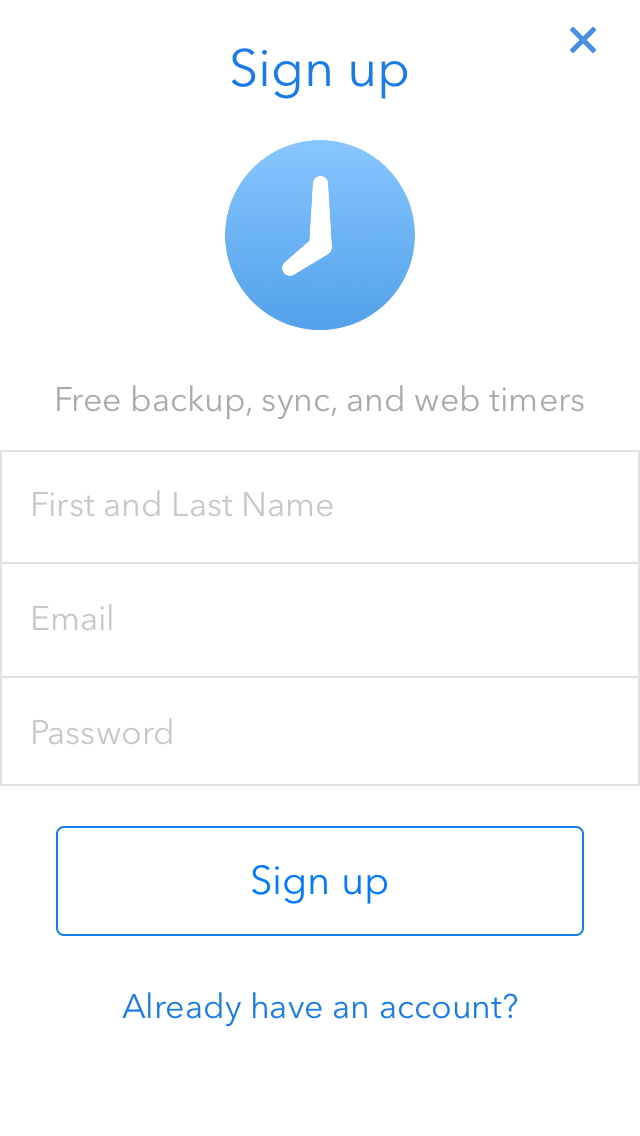
2. Utilities view

The utilities view includes a header and bars of utilities.



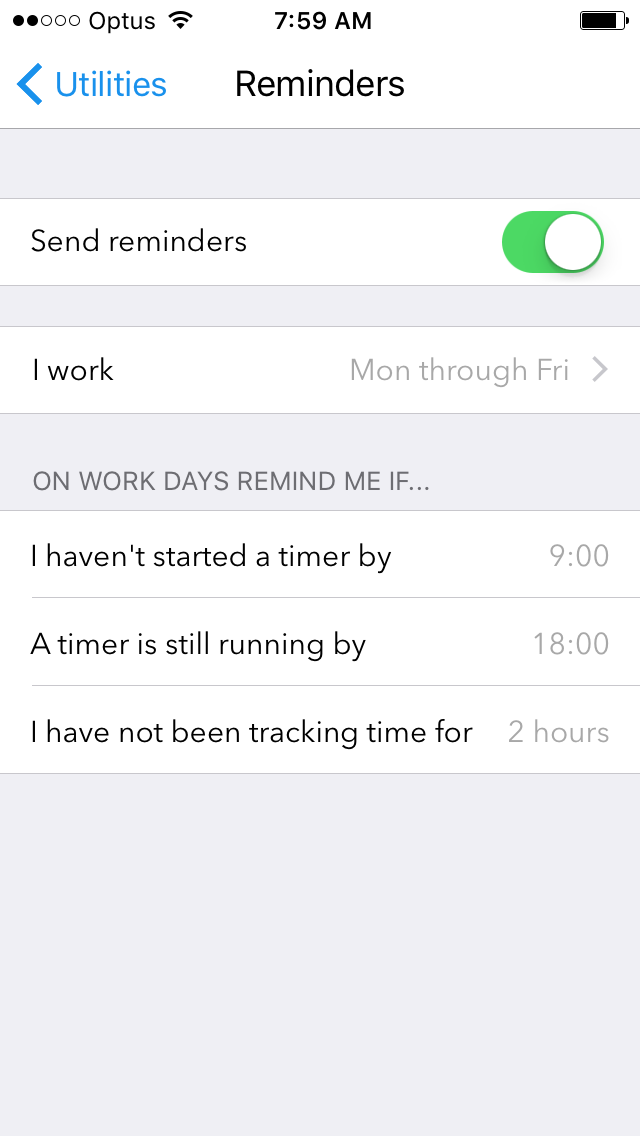
3. Sign up view

The sign up view includes the username, email address and password, and also with ‘Already have an account’ function.



4. Reminder setting view

The reminder setting view has a switch to turn on/off the reminder, and user can also set the weekday, duration and frequency of reminder.



### 2.2.2 Functionality

This app would allow users to identify the types of activities to track, and allow these to be ‘dragged and dropped’ into a timeline. Activities can be both resized and positioned by touch. User data will be sent to database, and database will save the data, then copy it to cloud. Besides, reminder function would be provided for user to set and resize the length of time for each activity.

On the other hand, the back-end interface would need to database the data, and allow export for subsequent analysis and visualization over a selected period of time.

1. Similar Functions:

* Sign up and sign in functions

Allow new user to sign up, and provide sign in function. User data will be sent to the database.

* Account settings (User profile)

Allow user to change their username and passwords using their old password and email address. User also can sign out.

* Utilities page which includes the settings (Sound, Hint, Rounding Rules)

User can turn on/off sound, as well as hints. The rounding rules enable time blocks snap to the nearest X minute increment when user adjust the start and end times of activities (x is 10 - 60).

* Allow user to add activities

User can add/delete the activities.

* Reminder (notification) function and settings

It has a switch to turn on/off the reminder, and user can also set the weekday, duration and frequency of reminder.

* Reporting (auto-report and upload) function

Automatically sending reports to the database

* Database (user and report)

A back-end database to store both the user data and report data.

* Visualisation of the report over a selected period of time

Report can be generated and visualised on the back-end interface.

2. Different functions

* A timer to record the time of each activity
* A timeline view for user to record and track the time of activities

These two functions are not necessary for our project.

3. Functions to explore and achieve

* Store the activities into the activity list

Provides activity list function for user to store the activities.

* Provide default types of activities

Allow user to use default activities

* In the timeline view, activities can be positioned (position can be changed)

User can change the position of activities in the timeline.

* In the timeline view, activities can be resized by touch

User can resize the length of activities by touch.

* Allow user to drag and drop activities to the timeline view

User can drag and drop activities into the timeline.

## 2.3 Summary

According to our research and analysis on integrated planner and Hours, we found that it is necessary to search the details of drag motion for IOS applications. There would be some changes for our application: remove timer, store activities, provide default activities, positioned and resized by touch, and ‘drag and drop’ function.

# 3 Methodologies Overviews

This section will introduce some common methodologies in software development, for example, Structured Systems Analysis and Design Methodology (SSADM), Scrum, Object-Oriented Programming and so on. The following content will introduce these methodologies and analysis their advantages and disadvantages.

## 3.1 Structured Systems Analysis and Design Methodology (SSADM)

The SSADM is one of waterfall model software development. It is a structured methodology and is hard to use. Using skilfully, it can produce well-documented, accurate information systems (SQA, 2007). Diagram 1.1 shows the whole process in the system development life cycle. However, the SSADM methodology focus on the analysis and design phase during the waterfall model of the systems development lifecycle.

The SSADM has many advantages. Firstly, the SSADM can determine information system viability by using three techniques. Avison and Fitzgerald claim that the Structured Systems Analysis and Design Method (SSADM) relies on multiple techniques so it is an excellent method (Abdelrahman, 2015). These techniques SSADM depended on are:

1. Logical Data Modelling, which includes identifying, modelling and documenting the data requirements of the system being designed.

2. Data Flow Modelling, which stands for the information flow around a system -- how it is changed and saved and the source of information outside the system.

3. Entity Behaviour Modelling, which is the process of identifying, modelling and documenting the business events that have an impact on each entity and the sequence where these events happen.

With these three techniques and viewpoints, the model is more accurate and complete. Secondly, using the SSADM has less chance for misunderstanding. This is because that SSADM methodology require a deep analysis of system at the beginning stages of the project. Besides, SSADM methodology is a mature approach of software development. Many developers are familiar with it. Therefore, developers will make less mistakes by using the SSADM. Thirdly, the SSADM methodology has rigid control on projects. Because the SSADM is a well-defined method of creating information system, every aspect of development process will be strictly under control. As a result, error hardly occurs during implementing. However, this feature can be seemed as a drawback. It is inevitable that the requirements for the system will change. Once the requirements changed, all the SSADM analysis may not work anymore.

The SSADM also has some disadvantages. The biggest drawback is that the SSADM is time-consuming. Because it takes too much time on analysis, there will be a large gap between the initiation of project and the delivery of system. If project manager does not allocate time to each phase rationally, the project will not be delivered on time. What is more, the SSADM is a great methodology for trained employees. However, if the company employ people who are not familiar with the SSADM techniques, the company may spend much more time and money on training novices.

## 3.2 Scrum

“Scrum is a project management methodology for agile software development that uses iteration and incrementation.” (Lei, 2017) Scrum was designed for projects which requirements may frequently change. The progress of Scrum is shown in diagram 1.2. It is acknowledged that the waterfall model is driven by documentation. During the whole development process, numerous documentation need to be written, and developer only implement the requirement on documents. However, Scrum focus on the face-to-face communication among people. There are eight phases in Scrum:

1. Product owner list product requirements, and this list is called Product Backlog

2. Scrum team, who is in charge of developing, estimate the workload and divide it into sprints (a sprint is an iteration and each iteration lasts one to four weeks)

3. With Product Backlog, Scrum team need to have a Sprint Planning Meeting and pick up a Story (one to four weeks’ work) as the work of current iteration. Then they will detail this Story, and the detailed Story is called a Sprint Backlog.

4. Scrum team will work on the Sprint Backlog. Every member in Scrum team will be allocated with a two-day task.

5. During every iteration, Scrum team will have a Daily Scrum Meeting. This meeting is aimed to understand the process of project. Every member in Scrum team must report what he/she have done and what he/she prepare to do. This meeting usually last fifteen minutes, and update own Sprint burn down (a diagram shows personal process). In addition, Scrum team can share their experience with teammates during the meeting.

6. Scrum require daily integration, which means there is a new vision of product. Once code pass the test, it should be updated instantly.

7. When a Story is finished, Scrum team should have a Sprint Review Meeting. This meeting will demonstrate the product to the product owner and client. (This is the most important meeting)

8. Finally, there will be a Sprint Retrospective Meeting. Scrum team share their experience and the drawbacks of current product, and put these drawbacks into next iteration requirements.

Comparing to traditional software development methodologies, Scrum is more flexible to unpredictability. However, it is difficult to practical with Scrum without an outstanding Scrum Master, who is responsible for managing and organizing the entire Scrum team.

## 3.3 Object-Oriented Programming (OOP)

Object-Oriented Programming is a type of computer programming that focus on objects, which are bundles of codes (Mohn, 2015). There are three concepts (class, inheritance, interface) about OOP will be introduced.

Firstly, “In OOP, a class is a template definition of the method s and variable s in a particular kind of object. Thus, an object is a specific instance of a class; it contains real values instead of variables.” (Rouse, 2005) Diagram 1.3 shows the relationship between objects, properties and methods. For instance, CAR is a class. Every car has speed, so speed is an attribute of CAR. Cars can move or stop. These actions are called behaviours of CAR. Therefore, developers can use bundles of codes to define what is CAR.

Secondly, a class can have subclasses, which reveal the idea of inheritance. In OOP, new objects can take on the properties of other objects (Yaiser, 2012). The original class is called a superclass or parent. The new object is called a subclass or child. Visible properties and methods of a parent are inherited by a child while additional properties and methods are added as well.

For example, CAR is the parent class. Developers can create a subclass named BUS. BUS will inherit the attributes and behaviours of CAR. In addition, BUS can have special attributes and behaviours, like routine. Besides, developers can write codes to override the behaviours of CAR or add new methods into BUS.

Finally, “An interface is a programming structure/syntax that allows the computer to enforce certain properties on an object (class).” A class is definition of a type of objects. An interface is a gate for instances to access the properties and methods of its own class. Interfaces reveal the idea of encapsulation in OOP. For example, when people want to launch a car, they must use a unique key. Interfaces are just like a “key” to properties in classes.

The advantages of OOP are obvious. Firstly, OOP can improve software-development productivity. This is due to three factors: modularity, extensibility and reusability. Secondly, because of these three factors, software is much easier to maintain as well. Thirdly, development lifecycle is shorter as reusing codes make development faster. Finally, reusing codes can also save money.

There are some disadvantages of OOP as well. Firstly, Object-Oriented Programming is difficult for some people. It takes time to get used with OOP. Secondly, OOP usually involve more codes which make the program larger. Thirdly, the program using OOP runs slower than procedural program because it has more instruction to be executed. Finally, OOP is not suitable for all types of program. Some programs may be much more complex by using OOP.

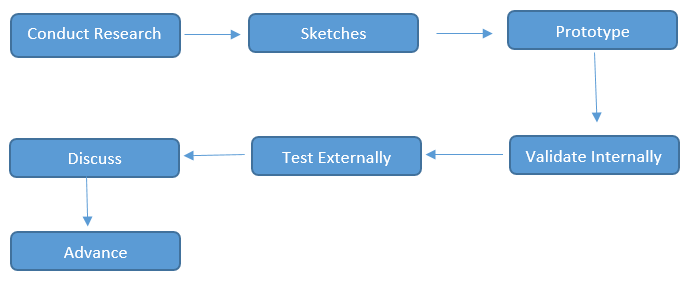
## 3.4 Lean UX

Lean UX is the practice of bringing the true nature of our work to light faster, with less emphasis on deliverables and greater focus on the actual experience being designed.

Lean UX is also a strategy that perfectly suits our vision and way of thinking. It focuses on efficiency and quality, thus aiming to improve the intended experience of the user.

Here is the Lean UX process:

Conduct Research -> Sketches -> Prototype -> Validate Internally -> Test Externally -> Discuss -> Advance



***Graph 2.0 Lean UX process (Gothelf, 2011)***

The Process aims to narrow down tasks and responsibilities into certain stages to ensure all are completed with equal importance, whilst upholding the aim of designing the actual user experience.

## 3.5 Methodology choice and justification

Our team choose to adopt Scrum and Object-Oriented Programming. As mentioned above, Scrum is more flexible than other software development methodologies. It is suitable for this project as client did not come up with detailed requirements. In addition, Scrum focus on the communication between people. It is good for application developers and UI designers to synchronize. Besides, all the developers are students so it is likely to have meeting daily which Scrum requires. They can also share more developing experience during daily meeting. However, they should pay attention to manage their project because it is dangerous to use Scrum without rigid management. As for OOP, it is appropriate for this project. This project requires to use SWIFT, which is an Object-Oriented language. Therefore, Scrum and Object-Oriented Programming are the ultimate methodologies for this project.

# 4 Technologies and Resources Overview

For doing this project, there are some of the technic is required to finish the task of each member. The Resources could be divided as 4 major aspects, planning phase, development phase, testing phase and launch phase. And this 4 major should have follow sub-skills.

## 4.1 Planning phase

Planning should involve each crew member. First of all, the project manager should provide a mature global plan, include tasking, time management, technology support and all the resource required. Than according to the global plan, there should be some sub-plans to support this project and have more details and all the activities that we should do in the whole project. Thus the project manage should using its organization skills to finish this task.

## 4.2 Development phase

This phase could be divided into 3 sub-phase. Front-end development, back-end development and database development.

For the front-end development will be assigned for the UI/UX designer. UI/UX designer should have enough graph skills. And this person should able to be good at storyboard and swift. Also the designer intend to use the Xcode to design this app. Then, the designer maybe should often communicate with back-end developers to know what kind of function be add or revise. Finally the designer should keep the whole interface clean and easy to handle for user.

On the other hand, back-end development will be assign to programmer or developer. They may use Apple developing language such as Swift and Cocopads. And the programmer should be able to using cloud space to save users’ data and allow user can get them activity data on different device. Also programmer should able to communicating with designer frequently to make this app more excellent.

Than is database development, this sub-phase requires the programmer should be able to using MySQL or Oracle expertly. Also the database developer should make sure the database can save users’ data and keep all of these data are on tap.

Finally, in this development phase, all programmer and designer should able using GIT to control the version of this project, the GIT also could be using as the backup technic. So we should choose GitHub to finish this task.

## 4.3 Launch phase

This phase talk about the environment builds. Developer should be using SSH to remote setting the environment of the server side development. The Linux skill is required for this phase. The developer should clear know the both front-end and back-end minimum models require.

## 4.4 Testing phase

Before we finish this project, we should test our app to find where we can improve and bugs in the app. There are some steps in this phase:

1. Doing a research or design questionnaire about the app, include the interface and easy use or not
2. Find 20~30 people who are in different age group, different gender and different career to test the time management application, let them note the problems they met during a one-week testing time, than fix this problems or bugs.
3. During the one-week testing time, the database programmer should check these users’ data every day to make sure the database can run successfully, in the meantime, back-end programmers should also check these data copy to the cloud or not.

# References

1. Kennedy, S. (2015). The 20 Best Time-Tracking Apps. [online] The Zapier Blog. Available at: <https://zapier.com/blog/best-time-tracking-apps>, Accessed 19 Nov. 2016.

2. Whipper, J. (2009). Blending sport and study — time management. *SPORTS COACH*, 30(4), pp.1-3.

3. SQA 2007, Outcome 1: Compare and Evaluate the Strengths and Weaknesses of Systems Development Methodologies, HN Computing, retrieved 2 December 2016 <[www.sqa.org.uk/e-learning/SDM01CD/page\_04.htm](http://www.sqa.org.uk/e-learning/SDM01CD/page_04.htm)>.

4. Abdelrahman, Oo 2015, 'Analysis of the University of Khartoum Library System Using the Structured Systems Analysis and Design Methodology (SSADM)', Cybrarians Journal, 38, pp. 1-24, Library & Information Science Source, EBSCOhost, viewed 1 December 2016.

5. Lei, H, Ganjeizadeh, F, Jayachandran, P, & Ozcan,P 2017, ‘A statistical analysis of the effects of Scrum and Kanban on software development projects’, Robotics & Computer-Integrated Manufacturing, 43, pp. 59-67, Applied Science & Technology Source, EBSCOhost, viewed 2December 2016.

6. Mohn, E 2015, 'Object-Oriented Programming (OOP)', Salem Press Encyclopedia Of Science, Research Starters, EBSCOhost, viewed 2 December 2016.

7. Rouse, M, 2005, Part of the Programming glossary, retrieved 3 December 2016, < http://whatis.techtarget.com/definition/class >.

8. Yaiser, M, 2012, Object-oriented programming concepts: Inheritance, retrieved 3 December 2016, < <http://www.adobe.com/devnet/actionscript/learning/oop-concepts/inheritance.html>>.

9. Interfaces in Object Oriented Programming Languages, n.d., viewed 3 December 2016, < http://www.cs.utah.edu/~germain/PPS/Topics/interfaces.html >.

10. Gothelf, J. (2011). Lean UX: Getting Out Of The Deliverables Business – Smashing Magazine. [online] Smashing Magazine. Available at: https://www.smashingmagazine.com/2011/03/lean-ux-getting-out-of-the-deliverables-business, Accessed 20 Nov. 2016.

# Appendix

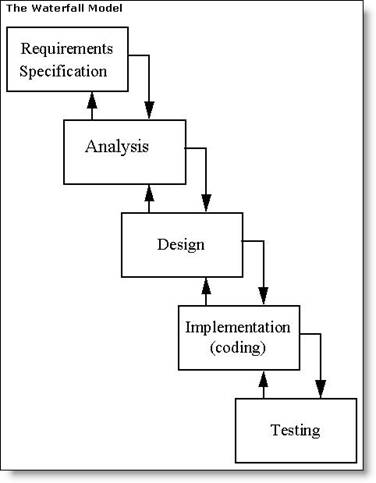


Diagram 1.1



Diagram 1.2

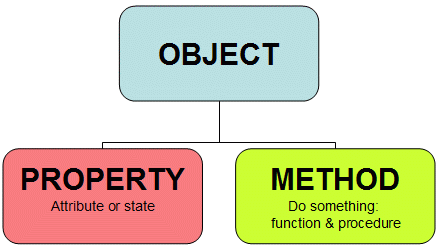


Diagram 1.3