

**Faculty of Arts and Sciences**

**Department of Computer Science**

CMPS 299 – Software Graduation Project

Spring 2017, Prof. Ahmad Dhaini

**Software Project Management Plan**

For the Group Term Project:

**TouristGo**

**LOGO**

*Team Members:*

Sirine Baba

Fawzi Chwayfaty

Mouhammad Al Houssein

Table of Contents

# Introduction

## Background

Tourism is one of the most important economic resources that all countries rely on. Many people have tourism is a favorite hobby. They like to visit touristic places in and outside their original country in order to get acquainted with new places, cultures, and traditions. Once cameras have been invented, taking pictures became an essential and necessary activity while visiting touristic places. You see all tourists carrying cameras and cellphones in order to take pictures and keep them as memories forever.

TouristGo is an android application that aims to help tourists to be guided to the best and most beautiful places around them. In addition, it gives them the opportunity to take pictures in every place they visit and share these pictures on social media, so their friends can interact with them and see touristic places that they did not know about before, so they can visit them too. All pictures that are taken will be saved in the gallery and kept as memories. Furthermore, all pictures that are shared on social media will be saved in the user’s profile inside the application, so he/she can reside to them anytime, anywhere.

## Business Opportunity

The problem being solved is the confusion of tourists when visiting new countries, villages, towns, or any new areas that they did not visit before. Based on a personal experience, and on many tourists’ experience, being lost in a place is the most annoying thing that can happen with a tourist who needs to benefit from his time in visiting beautiful places, especially is he/she has a limited time to have fun like on a weekend, or during a small vacation. Based on that, TouristGo was created in order to solve all these problems, and to help tourists benefit from their time to the maximum, enjoy their trips, and have a lot of fun.

## Customer or Market Needs

In the current market, there are a few applications or websites that touch this aspect of life, but tourists have needs that exceed the scope of what the aforementioned software have to offer, like getting the directions to the touristic place, getting a description about the place along with a picture, etc… As such, tourists will be able to use our software to directly get all necessary information about the touristic places that are around them, visit any place, and take pictures in order to keep them as memories, or share them with friends on social media.

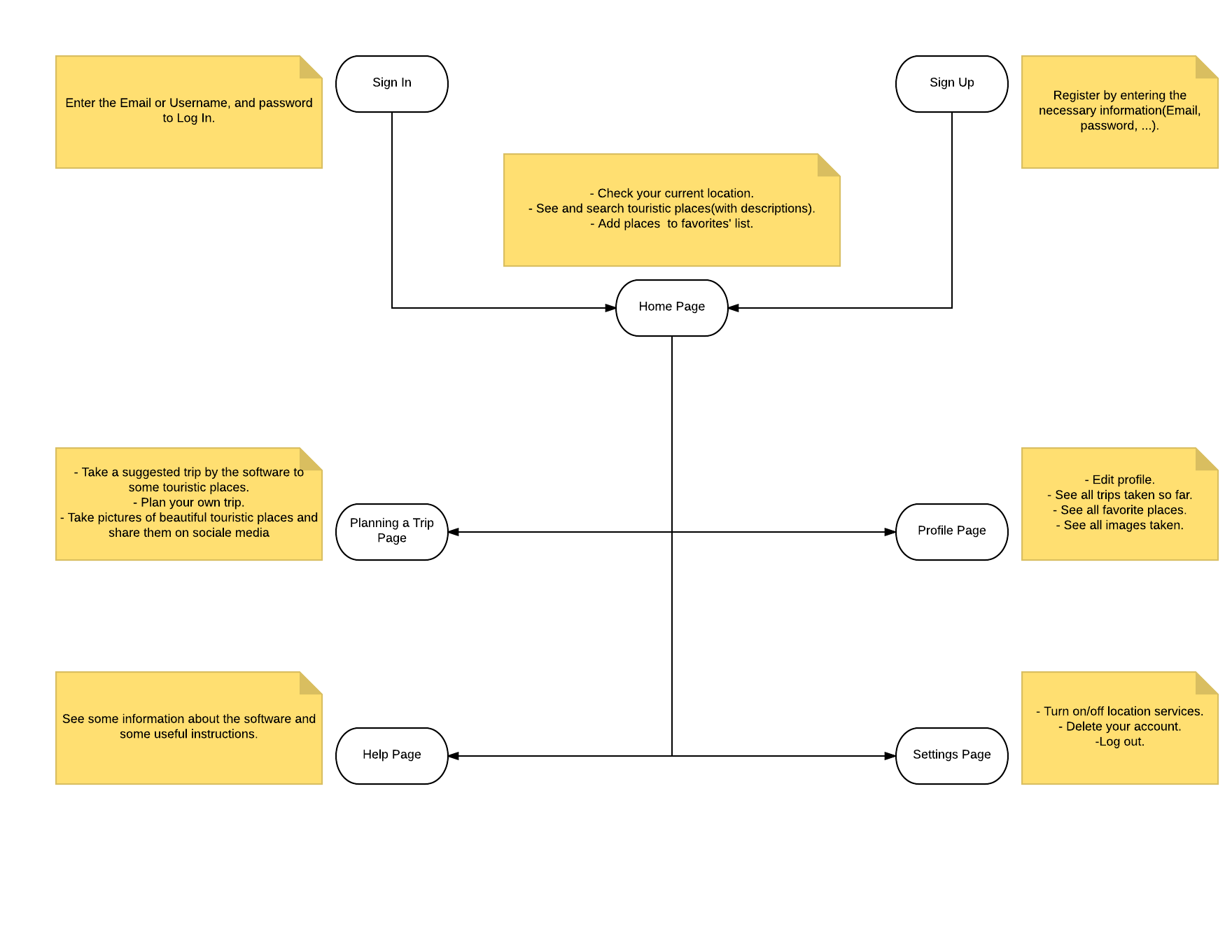
## Business Objectives and Success Criteria

* Positive feedback from tourists on our software.
* A good portion of the tourists has downloaded and is actively using the software (at least 500 tourists).
* Attracting potential advertisers.
* In future releases, the amount of money earned from advertisements will be our primary measure of success.

The faster we finish and release our software, the more likely it will be successful because the time to market is an important factor in developing software. In our specific case, there is the likely chance that there are other development teams working on a similar product due to the increasing demand of a software that caters to the needs that our software targets. The sooner we finish our software, the better the chance to get ahead of the competition, and guarantee that our software is the first to be used.

# Vision

## Story-Telling Diagram



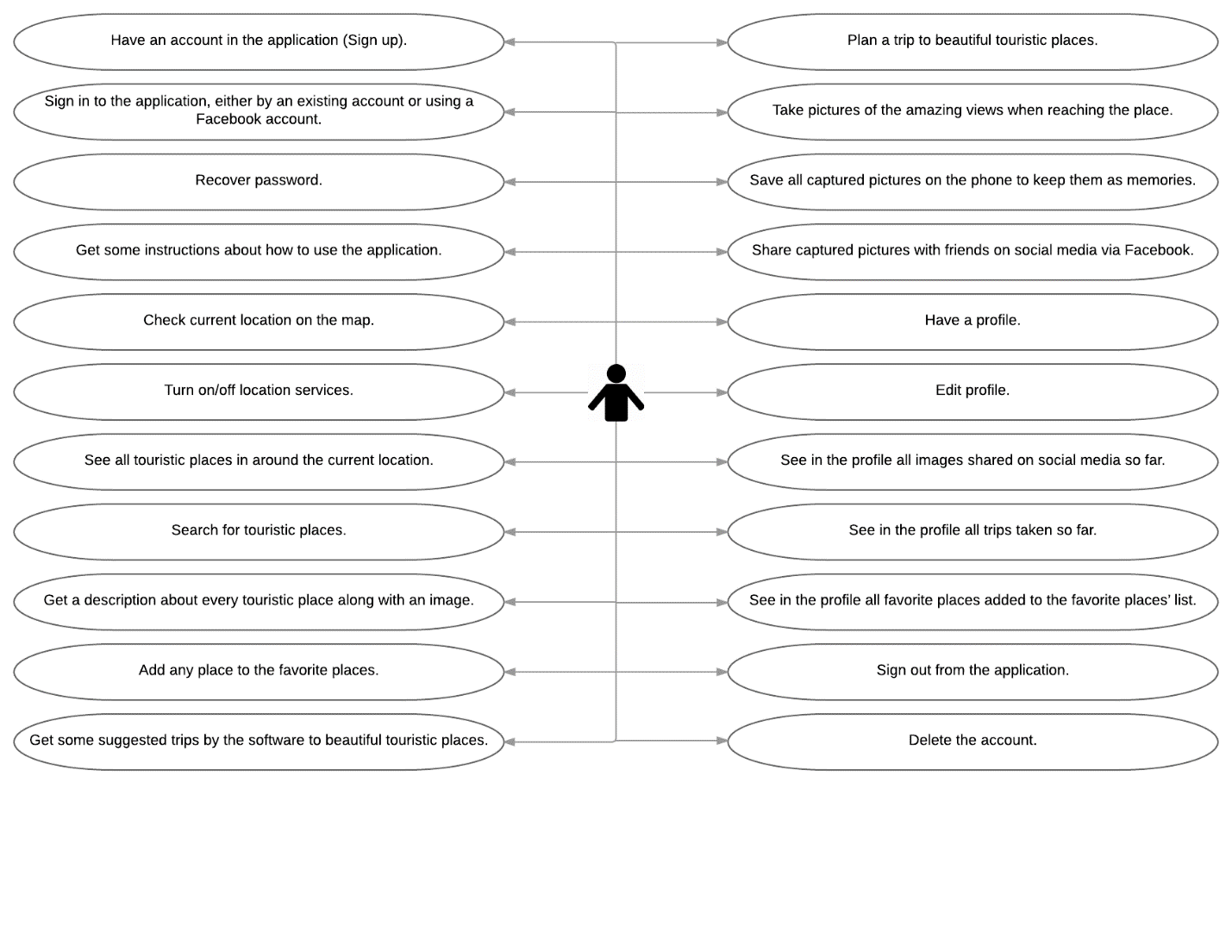
## 

## Major Features

1. See all near touristic places.
2. Search touristic places.
3. Get a description about every place along with a picture.
4. Add places to favorites.
5. Take a trip to many touristic places suggested by the software.
6. Plan a trip to touristic places based on users’ choices.
7. Take pictures of beautiful views.
8. Save these pictures in the cellphone’s gallery.
9. Share these pictures on social media.
10. Get a list of pictures shared on social media inside the user’s profile.
11. Get a list of favorite places inside the user’s profile.
12. Get a list of all trips taken by the user inside his profile.

# Requirements

## Use Case Diagram



## User Stories

|  |
| --- |
| As a user, I would like to have an account in the application (Sign up). |
| As a user, I would like to sign in to the application, either by an existing account or using my Facebook account. |
| As a user, I would like to recover my password in case I forgot it. |
| As a user, I would like to get some instructions about how to use the application. |
| As a user, I would like to check my current location on the map. |
| As a user, I would like to turn on/off location services. |
| As a user, I would like to see all touristic places around me. |
| As a user, I would like to search for touristic places. |
| As a user, I would like to get a description about every touristic place along with an image. |
| As a user, I would like to add any place that I liked to my favorite places. |
| As a user, I would like to get some suggested trips by the software to beautiful touristic places. |
| As a user, I would like to plan my own trip to beautiful touristic places. |
| As a user, I would like to take pictures of the amazing views when I reach the place (access the camera from the application). |
| As a user, I would like to save all captured pictures on my phone to keep them as memories. |
| As a user, I would like to share captured pictures with friends on social media via Facebook. |
| As a user, I would like to have a profile. |
| As a user, I would like to edit my profile. |
| As a user, I would like to see in my profile all images shared on social media so far. |
| As a user, I would like to see in my profile all trips taken so far. |
| As a user, I would like to see in my profile all favorite places that I have added to my favorite places’ list. |
| As a user, I would like to sign out from the application. |
| As a user, I would like to delete my account in case I do not want to use the application anymore. |

## Screen Mockups (not done yet)

## Non-Functional Requirements

* Data: data will be stored and available to users on the server for up to 5 years, after which we will dispose of it.
* Backup: stored data will be updated in the databases any time a user commits a change to their account.
* Price: our software will initially be open source – we have no current intention to make our users pay for our software.
* Localization: our software will be primarily functioning in Lebanon, and will consequently be adhering to the country’s rules, regulations, and policies concerning software distribution.
* Documentation: any changes to our software (updates, new features, bug fixes, etc.) will be recorded and documented in our SPMP.
* Extensibility: our software has a lot of potential for adding new features beyond what our initial release will encompass (which are listed in section 4.2), all of which will help in keeping our users interested in our software and in helping tourists get the most out of our software.
* Resource constraints: our system requirements are minimal, so we can expect anyone with a working android device and an active internet connection to benefit from our software.

# Managerial Process and Project Organization

## Management Objectives and Priorities

Our main priority is to have our software up and running and ready for release as of Thursday, May 18th, 2017. Our emphasis will be on making sure that all the key features of the software – which will be primary appeal to our consumers – get implemented. Moreover, we will be focusing more on quality considering the amount of time we have to develop the project, and the relative ease of actually implementing these key features.

The main objectives we wish to accomplish is to get a good portion of the Tourists to actively use our software and get positive feedback from them. On the long run, we hope to attract potential advertisers after we have obtained a considerable number of users.

## Assumptions, Dependencies, and Constraints

We assume that tourists will not try to attack nor corrupt our software or any of its data, but as a precaution the appropriate countermeasures will be implemented to deal with such security issues in future releases provided that our software gains enough popularity. Moreover, our most critical assumption is that we will be able to get as much touristic places as possible in near countries, and after that in the Middle East.

The constraints we will be essentially dealing with is the amount of time we have to complete in the development of our software. As a result, some “great” or “nice to have” features may be excluded from the initial release. Despite our confidence that we will be able to implement the features we have initially set, the time constraint is still an issue that must be taken into account.

## Project Risks

### Discussion of Risks to be managed

Due to the arguably controlled environment that we are working with, there is small chance of any potentially harmful occurring to our project. However, some complications may arise from within the team, or from some unpredictable factor. Consequently, our risks can generally be classified into three categories: 1) known, 2) predictable, and 3) unpredictable risks.

1. Known risks are risks that we know we are most likely to be faced with due to careful considerations of the factors we are working under during the development of our plan. Risks in this category include: time pressure or not having enough time to complete the program, and getting stuck on a problem in coding that our developers cannot solve.
2. Predictable risks are risks that we know, from experience, are likely to occur. Risks in this category will include: conflicts among team members on certain issues, a member experiencing too much work load from other courses, change in a major requirement, and the addition of a new major requirement that needs to be implemented into the software, and users ultimately not liking the product.
3. Unpredictable risks are risks that we cannot identify its occurrence in advance. Risks in this category include: hard drive failure i.e. losing our program code or having it deleted, a team member getting sick, potential political instability in Lebanon leading to riots in the streets, and prolonged power outages in Lebanon.

#### Risk Mitigation

The primary way we can mitigate most of the risks mentioned above that are categorized as “known” and “predictable”. Planning ahead helps in knowing how to set realistic time goals (risk 1 and risk 4), it helps in identifying ahead of time where the most difficult parts of the code may be (risk 2), it helps in keeping all team members clear on objectives and keeps everyone on the same page (risk 3)

As for “unpredictable” risks, it is very difficult to foresee any of this risks occurring, but luckily we have adequate management controls to take care of the damage these risk may cause even in the unlikely chance that they do occur.

#### Risk Monitoring

In order to best monitor our project and foresee the occurrence of any possible risk, communication will be key. Excluding the “unpredictable” risks, we can get a fairly good idea of when we may fall victim to one of the risks mentioned above if team members communicate well with one another. We can know ahead of time if we are behind on schedule (risk 1), or if the developers have been stuck on a part of code for while (risk 2). Also, we can through constant communication with our users know if there will be a change or an addition of any requirements that the users consider critical and thus must be implemented in the initial version of the software.

#### Risk Management

The general way that we will be managing any risks is to temporarily shuffle around the responsibilities between team members. However, the way we will specifically manage each risk is located in the tables below in section 10.4.2.

## Monitoring and Controlling Mechanisms

Team meetings are conducted on a weekly basis. In these meetings we discuss what’s been done since last meeting, what needs to be done by next meeting, and any specific problems that a team member(s) may be facing. Team members are also in constant communication via WhatsApp so that nothing important may go unknown.

In addition, all files and documents that are used/created at any point or for any reason during development are stored in a shared folder on Dropbox so that all team members may have access to them and know what’s going on.

Our team is based on a system of constant checks and balances in which all team members monitor one another’s work and make sure that each members is doing what he is supposed to.

## Weekly Progress Reports (not done yet)

# Technical Process

## Methods, Tools, and Techniques

We will primarily be following an Agile methodology for our **SDLC**, but will also be incorporating some aspects of plan driven methodology in order to better organize our time and work throughout the semester.

Consequently, our work will be split up into several sprints, and each sprint will be dedicated to working on certain requirements that the team agrees upon. After breaking down the work into these sprints, and before we begin actually coding any part of the sprint, we will work on creating a high level representation of how the requirements will be implemented in code. This will help in organizing and structuring the work for the corresponding sprint, and will also give us a better idea of what we need to implement the code (i.e. any helper methods, specific functions that can be used, etc.).

Below are listed the main tools we used throughout the project:

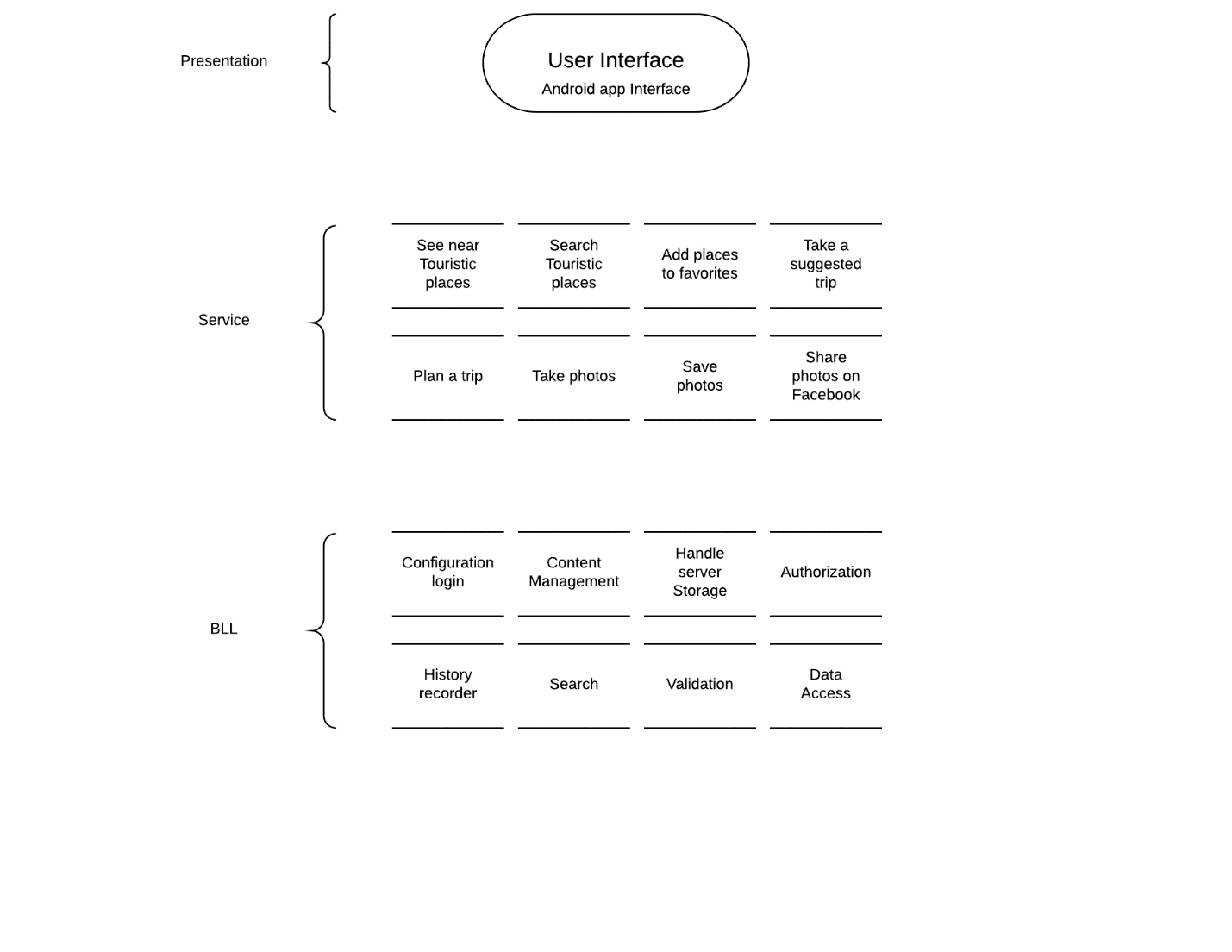
* Android Studio IDE for writing the code and developing the software.
* Firebase to create the necessary databases which will be integrated into the code written in Android Studio.
* Balsamiq Mockups to create mock ups of the screens of our software.
* Realtime Board present/display the application flow.
* WhatsApp primarily for communication between team members.
* Microsoft Word to create the **SPMP.**
* Lucidchart which is a Website to create diagrams in the SPMP.
* Microsoft PowerPoint to create the final presentation.
* Dropbox to store and backup any important files for the project, and also for easily sharing files between team members.

## Software Documentation

### Design Document

Our program will be primarily written in Android Studio. As for our database, we will be writing it in Firebase and connecting it to our code in Android Studio. Ultimately, our program will be running primarily as an android application. After the application has been created, it will then be uploaded to the Play Store for tourists to download.

### Technology Stack



### Database Diagram(not done yet)

### User Guide

1. Download the application from Play Store.
2. When you open the application, you will be directed to a Sign in page.
3. If you don’t have an account, click on Sign up to create one (if you have an account, move on to step 6).
4. Fill in the necessary field to create an account.
5. After creating the account, you will be directed to the Home page.
6. From the Home Page you can check your current location, see and search touristic places along with descriptions, and add places to you favorites’ list.
7. In the Trips page, you will be able to take trips suggested by the application. Also, you will be able to plan your own trip to any desirable touristic place.
8. In the profile page, you will be able to see all trips that you have taken so far, all your favorite places, and all your images that you have shared on social media through the application.
9. In the help page, you will get some instructions on how to use the application.
10. In the settings page, you will be able to turn on/off your location service, delete your account, and log out from the application.
11. And finally, enjoy! :D

### Hosting Environment Configuration

No third-party will be brought in to manage our data nor run our programs, our developers will be hosting the program themselves, and will be keeping track of the software’s database and integrity.

## Project Support Functions

* Ensure the agreed project management methods, standards and processes are maintained throughout the project lifecycle.
* Assist in the production and maintenance of project plans.
* Develop and implement appropriate configuration management procedures
* Co-ordinate the production of all reports and produce project summary reports.
* Define and document procedures in accordance with agreed methodology.
* Maintain risk and issue logs and change control records.
* Develop and support effective communication mechanisms between the project team members.

## Configuration Management

We are recording weekly meetings to ensure that the reason and origin of any decision that is made and/or change agreed upon is well documented in case for the need for the sake of traceability.

## Quality Assurance and Control

In order to make sure that our software is as error free as possible, we will be following several guidelines to ensure that the software’s quality is up to par. To begin with, our developers will be following the “pair programming” technique where more than one person will be working on simultaneously on the code, making it less likely that that a bug or error goes by unnoticed. In addition, we will be handing our code to many testers who will subject it to rigorous testing with their goal being to break the code and uncover any of the software’s weak spots. This testing will be done in iteration, each sprint will lead to subsequent parts of the code being completed, and therefore each part of the code will be tested independently, so that if any errors or bugs occur after adding more parts of the code we can better approximate where our errors would be (most probably from the newest parts of the code).

# Work Packages, Schedule

Concerning our schedule, we will supposedly by the deadline set by the professor (Thursday, May 18th, 2017) have fully completed the software including all the required documentation (mainly the **SPMP)**. Upon completion, the source code will be sent to the professor for inspection/grading, and the **SPMP** will be printed and also handed to the professor.

# Project Resources

## People

* Ahmad Dhaini:
  + Professor
  + [ad57@aub.edu.lb](mailto:ad57@aub.edu.lb)
* Sirine Baba:
  + Team member
  + snb15@mail.aub.edu
* Fawzi Chwayfaty:
  + Team member
  + [fjc03@aub.edu.lb](mailto:fjc03@aub.edu.lb)
* Mouhammad Al Houssein:
  + Team member
  + [maa244@aub.edu.lb](mailto:maa244@aub.edu.lb)

## Hardware and Software

No special hardware was needed for this project besides a computer with the latest version of Android Studio installed on it, and with a stable internet connection.

## Special Resources

No special resources were used for this project.

## References

Professor Ahmad Dhaini was our primary reference for anything needed to be made for the project, specifically any documentation.

For any other needed information, we had the YouTube and all of the Internet at our disposal.