**Consilium**

**Functional Requirements Document**

**The Daily Planner App**

**Jonathan Vargas**

**Tim Pasion**

**Gurinder Brar**

**Xia Hua**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Description of Change** | **Author** | **Date** |
| 1.0.0 | Functional requirements created |  | 10/21/2019 |
|  |  |  |  |

CONTENTS

1 INTRODUCTION 4

1.1 Purpose 4

1.2 Scope 4

1.3 Background 4

1.4 References 4

1.5 Assumptions and Constraints 4

1.6 Document Overview 5

2 METHODOLOGY 5

3 FUNCTIONAL REQUIREMENTS 5

4.1 Context 5

4.2 User Requirements 5

4.3 Data Flow Diagrams 6

4.4 Logical Data Model/Data Dictionary 6

4.5 Functional Requirements 6

5 OTHER REQUIREMENTS 6

5.1 Interface Requirements 6

5.2 Data Conversion Requirements 7

5.3 Hardware/Software Requirements 7

5.4 Operational Requirements 7

APPENDIX A - GLOSSARY 11

1. **INTRODUCTION**

Procrastination and bad time management are the two most detrimental habits that inhibits or slows down your potential of achieving a successful life. The Consilium mobile application will enable you to add events and keep track of them by being notified. This will allow you to achieve maximum productivity daily and help develop better habits that will enhance your life.

## Purpose

The purpose of this document is to allow the stakeholder(s) to have an overview of the software’s usability requirements for full functionality and its utility that it will provide.

The Consilium mobile application will provide the user with a planning experience that will boost their productivity and time management skills. Through a To-do/Task orientated environment.

## Scope

This document deals with a consilium mobile planner application, this document ensures our project’s scope is accurately defined and mapped. The detailed set of deliverables are derived from the project and these deliverables are derived from project’s requirement. The processes used to accomplish its purpose were planning, controlling and finalizing ideas.

## Background

We are team Consilium. It consists of Gurinder Singh, Xia Hua, Jonathan Vargas and Tim Pasion.Our responsibility is to develop a daily planner app or minimum viable product (MVP) by the deadline specified within our semester of ENSE 374. As such, we have decided on creating our proposed mobile app. Currently, we have planned to split the work on this functional requirement document as there are Four of us and this document is long and would help to finish on time.

This document is being produced by the development team for the stakeholders/users that will be using the Consilium mobile application. This document is being produced to outline the method of designing this application and its limitations that will inhibit this application from functioning. This document will also outline the requirements for safe and friendly experience by having use cases.

## References

Github, Google Drive, Discord

### Assumptions

1. Examples of assumptions include: availability of a technical platform, legal changes and policy decisions.

1. Design/requirement changes to the applications that may happen. Delay deployment.
2. Software and frameworks functionality may stop working. This will cause the application to be only on either iOS or Android platform rather than cross platform.

### Constraints

Constraints are boundary conditions on how the system must be designed and constructed. Examples include: legal requirements, technical standards, strategic decisions.

* Constraints exist because of real business conditions. For example, a delivery date is a constraint only if there are real business consequences that will happen as a result of not meeting the date. If failing to have the subject application operational by the specified date places the organization in legal default, the date is a constraint.
* Preferences are arbitrary. For example, a date chosen arbitrarily is a preference. Preferences, if included in the FRD, should be noted as such.]
* Some degree of an MVP must be achieved by the end of our semester. Time constraint. Failure or low quality results by the deadline could result in bugs .
* Time constraint for completion. The overall time to complete this project from design,coding, and testing. Not having full functionality of functions other than the main ones.
* Knowledge of code. Using different coding languages and frameworks to develop the application.
* Resources: In terms of the amount of people writing clean and efficient code. This will also influence the time constraint.

## Document Overview

The contents of this documentation are intended to explain the current arranged plan, needs, limitations and safety measures for the undertaking. We will examine how we find functional requirements, what they are, the manner by which our app structure will frame to oblige them, and how these functions should work. We will likewise detail the significance and core values behind our applications security, dependability, recovery just as other general needs like software requirements At long last, we will likewise dig into progressively broad prerequisites like error handling.

1. **METHODOLOGY**

This document fills in as a constantly changing framework of our application's arranged design. It will advance as we keep on building up our application and grow better knowledge of what we are expected to make. This document will be changed as we reach, complete, or re-plan achievements dependent on fluctuating advancement obstacles. As of now, the report traces our idealized design, or how we're working on it. This report will be a reference record as we plan our application.

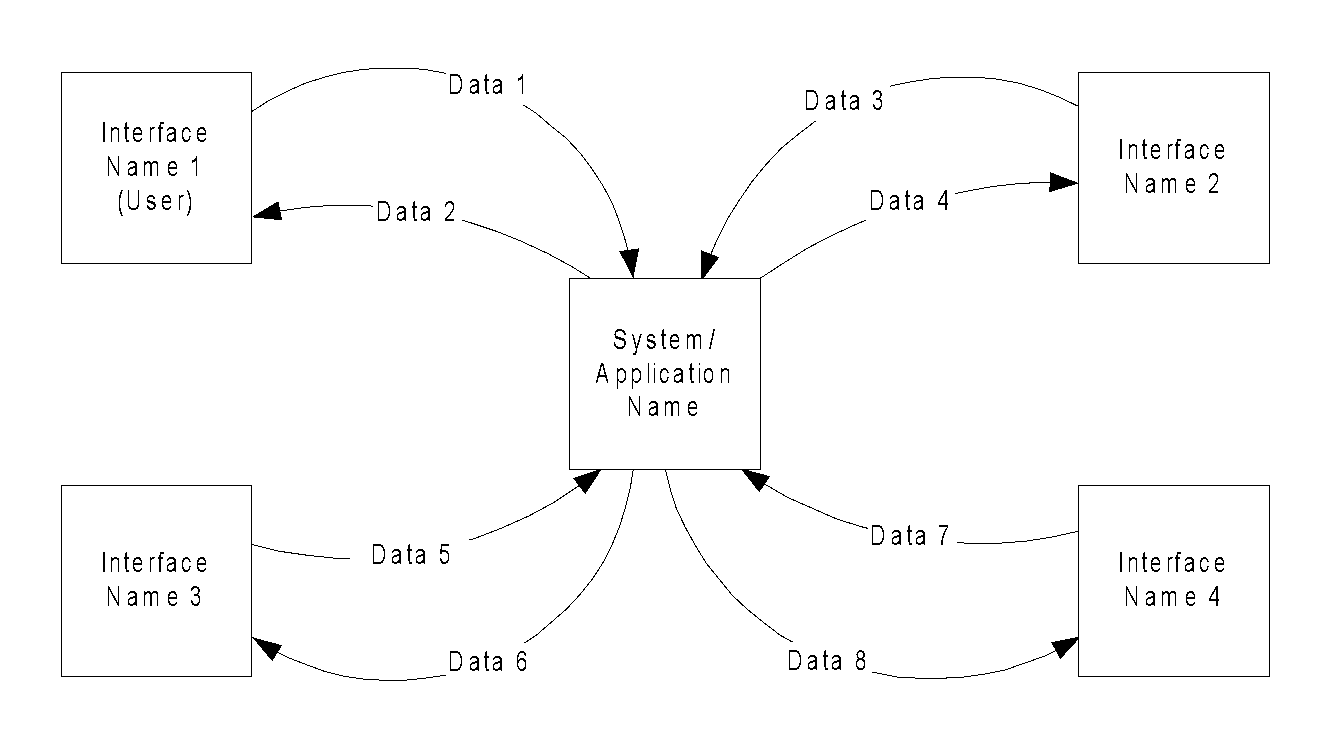
Necessities might be found as we proceed to create and find out about the idea of this sort of use.

1. **FUNCTIONAL REQUIREMENTS**

## Context

[Provide a context diagram of the system, with explanations as applicable. The context of a system refers to the connections and relationships between the system and its environment.]

Exhibit 2 - Generic Context Diagram

**

## User Requirements

A user of Consilium is a person that has busy days and needs something to keep track of their activities. Examples of users are: Students, Freelancers, and managers. A student may want to use the To-do list as a way to track what dealines to prioritize. A freelancer may use Consilium as a way to schedule their jobs and projects. A manager may use the calendar function to schedule meetings and may use the To-do list to track what needs to be prepared. All users will only be able to manipulate their own events/to-do lists.

A user can expect to enter their basic information such as their username and email to be given their calendar. A user can input a task or event and be expected to view their event in the calendar or to-do interface.

## Data Flow Diagrams

[Decompose the context level diagrams to determine the functional requirements. Data flow diagrams should be decomposed down to the functional primitive level. These diagrams are further decomposed during design.]

## Logical Data Model/Data Dictionary

[Create the initial Logical Data Model. Describe data requirements by providing data entities, decomposition, and definitions in a data dictionary. The data requirements describe the business data needed by the application system. Data requirements do not describe the physical database and are not at the level of identifying field names.]

## Functional Requirements

[List the functional requirements of the system.]

Keeping the students on track with daily reminders on the activities performed in campus

Users can add ToDos/events

Users can view ToDos/events

### Functional Requirements Group 1

[List the functional requirements for each functional requirements group.]

Exhibit 4 - Sample Requirements Group 1

|  |  |
| --- | --- |
| **Section/ Requirement ID** | **Requirement Definition** |
| FR1.0. | The system shall [parent requirement group 1]. |
| FR1.1 | The system shall [child/parent requirement]. |
| FR1.1.1 | The system shall [child requirement]. |
| FR1.1.2 | The system shall [child requirement]. |

### Functional Requirements Group 2, Etc.

1. **OTHER REQUIREMENTS**

[Describe the non-behavioral requirements.]

## Interface Requirements

The user interface that will be implemented are the list of *To Do’s / Events , Calendar , User Profile*. User can interact with the interface by adding the event thus the modal can update the view on the *To Do’s and Calendar* interfaces. The *User profile* will give the user the customizability of the overall UI .

### Hardware Interfaces

cell phones and PCs should have the option to interface with our application through suitable web browsers. Our application will in a perfect mode to fit in the changing screen sizes. This mobile application will be cross platform, thus will be available for iOS and Android users.

### Software Interfaces

Our application will interface with the firebase as we using that for our database.

### Communications Interfaces

We will need to have access on students timetable and schedule, this may involve their communication interfaces. Possibility: Communicate with UR self service.

## Data Conversion Requirements

To have the user to agree to display their email and name onto the app.

## Hardware/Software Requirements

To make this application functional for both iOS and Android this will be developed using Ionic Framework , Cordova , Angular JS, CSS. This cross platform application can access some of the mobile devices built in functions such as calendar, camera, etc.. . Cordova allows us to develop applications with HTML5, CSS, and javascript. Though we will be using AngularJS to make the application more functional and design wise better. Backend we will be using MySQL and PHP.

## Operational Requirements

[Provide the operational requirements in this section.

Do not state how these requirements will be satisfied. For example, in the Reliability section, answer the question, “How reliable must the system be”? Do not state what steps will be taken to provide reliability.

Distinguish preferences from requirements. Requirements are based on business needs, preferences are not. If, for example, the user requires a special response but does not have a business-related reason for it, that requirement is a preference.

Other applicable requirements on system attributes may be added to the list of subsections below.]

Operational requirements describe how the system will run and communicate with operations personnel.

### Security and Privacy

A. State the consequences of the following breaches of security in the subject application:

1. Loss or corruption of data

Loss or corruption of data will clear their future events, thus nothing will show in the calendar or the list of events.Also may lose account information.

1. Disclosure of secrets or sensitive information

This application will only require the sensitive information of the user’s email address, thus the consequences may be spam sent to their email. If the user’s account is hacked then information about the user’s schedule will have the possibility of being leaked.

1. Disclosure of privileged/privacy information about individuals

Email and first name will be exposed

1. Corruption of software or introduction of malware, such as viruses

If this software is introduced to viruses or malware this will affect the user’s OS on their mobile device.

1. State the type(s) of security required. Include the need for the following as appropriate:
2. Physical security.

N/A

1. Access by user role or types.

To access this application users can sign up for this application. Signing up is not mandatory for operational use.

1. State access control requirements by data attribute. For example, one group of users has permission to view an attribute but not update it while another group of users has permissions to update or view it.

This application will be independent thus other users will not be able to interact with other user’s schedule.

1. State access requirements based on system function. For example, if there is a need to grant access to certain system functions to one group of users, but not to another. For example, "The system shall make Function X available to the System Administrator only". The functionality of this such as the *Add Event* , and *Calendar* can be accessed by user.
2. State if there is a need for certification and accreditation of the security measures adopted for this application] N/A

*The Security Section describes the need to control access to the data. This includes controlling who may view and alter application data.*

### Audit Trail

[List the activities recorded in the application’s audit trail. For each activity, list the data recorded.]

**Add\_Event** function will record the following data:

* Title of event
* description of event
* date-start and date-end
* tag
* priority
* location

**Sign up** function will record the following data:

* Name
* Email
* Password
* Confirm password

**edit Event** function will record the following data:

1. New title of event
2. New description of event
3. New date-start and date-end
4. New tag
5. New priority
6. New location

### Reliability

A. [State the following in this section:

1. State the damage can result from failure of this system—indicate the criticality of the software, such as:
2. Loss of human life N/A
3. Complete or partial loss of the ability to perform a mission-critical function

N/A

1. Loss of revenue

N/A

1. Loss of employee productivity

No damage to the mobile device or user if the system fails

1. What is the minimum acceptable level of reliability?

The minimum acceptable level of reliability is 0.7.

B. State required reliability:

1. Mean-Time-Between-Failure is the number of time units the system is operable before the first failure occurs.
2. Mean-Time-To-Failure is the number of time units before the system is operable divided by the number of failures during the time period.
3. Mean-Time-To-Repair is the number of time units required to perform system repair divided by the number of repairs during the time period.]

*Reliability is the probability that the system processes work correctly and completely without being aborted.*

### Recoverability

A. In the event the application is unavailable ```````````````````````````````````````````````````````````````````````````` ` to users (down) because of a system failure, how soon after the failure is detected must function be restored? If the system fails then the time frame for the system to be running again should be within an hour being the best case, the worst case would be a day.

B. In the event the database is corrupted, to what level of currency must it be restored? For example “The database must be capable of being restored to its condition of no more than 1 hour before the corruption occurred”.

If the database becomes corrupted then the previous version that the application can return would be a complete wipe since you can go back to previous version though events from the past will be there which will be a hindrance.

C. If the processing site (hardware, data, and onsite backup) is destroyed, how soon must the application be able to be restored?]

*Recoverability is the ability to restore function and data in the event of a failure.*Looking at our application it would not be too easy to restore our processing site. However, the need for restoration is urgent given the scope of this app.

### System Availability

Our app should be running 24/7. Downtime for maintenance and updates should be reserved for midnight.

Expected peak times: 8:00am - 10:00pm

System availability will be everyday 24 hours since this will allow the user to add events whenever they want and will be able to see the events on the calendar or the list display.

### General Performance

Required response time is unknown, but results as fast as possible.

Errors should almost immediately informed to the the user.

We expect more than ~1000 visits per day at peak, given the size of our university. This is assuming all students that are aware of our app frequently use it.

### Capacity

[List the required capacities and expected volumes of data in business terms. Do not state capacities in terms of system memory requirements or disk space—if growth trends or projections are available, provide the]

The users for this app are for university students at the U of R and will cater to every student if possible to scale larger with the resources we have. The capacity hopefully can be of general purpose use for everybody.

### Data Retention

[Describe the length of time various forms of data must be retained and the requirements for its destruction.

For example, “The system shall retain application information for 3 years”. Different forms of data include: system documentation, audit records, database records, access records.]

Data that will be retained are events the user adds. These will show until the user is able to validate that they have completed the task/event. These events will also expire when the date has passed (24 hours after the scheduled event )

### Error Handling

Error handling system will be red text over form inputs if they are required and are empty then the user will not be able to proceed with the operation. This will occur in the function of adding events to their To Do’s/Calendar display. Error handling will also occur in sign up and login forms where required fields must be field to continue , in turn linked to validation.

### Validation Rules

Validation of users’ login information:

Email and Passwords are required information to proceed to login.

Validation for Signing up

User has to enter their email, name ,password, and confirm password

Email will have to follow a specific format using Regex. Email cannot be null/empty and will have to contain ‘@’ sign.

Password will have to follow a specific format using Regex. One capital letter, minimum 8 characters long and a non-character.

Confirm password will have to match the password.

To *AddEvent* function the user will have to fill out the form given.

Title , description,tag ,date ,priority level,location.

### Conventions/Standards

[Describe system conventions and standards followed.

For example: Microsoft standards are followed for windows, Institute of Electrical and Electronics Engineers (IEEE) for data formats, etc.]

**APPENDIX A - GLOSSARY**

[Define terms, acronyms, and abbreviations used in the FRD.]

Regex: Regular Expressions

JS: Java script

UI:User interface

N/A: Not applicable

UR: University of Regina