Software Requirements Specification

for

SELECT AGENCY EMPLOYEE MANAGEMENT SYSTEM

Version 1.0 approved

Prepared by: Amine Mallah, Armand Ismail, Majid Badawi, Pierre Abi Chakra

CSC 490- Software Engineering: Submission II

3/18/2018

Table of Contents

0. PREFACE – PLEASE READ FIRST

0.1 PURPOSE OF THIS DOCUMENT

#1 *This document is a customised Requirement document for use SELECT Agency and Software Engineering class. It provides guidance and template material which is intended to assist the relevant management or technical staff, whether client or supplier, in producing a project-specific Requirement document. It is also useful background reading for anyone involved in developing or monitoring the implemented system.*

0.2 USE OF THIS DOCUMENT

#1 *This Preface is addressed to the users of this customised document.*

#2 *The remaining sections (numbered 1, 2, 3,...) constitute a template that should be used to construct the project-specific User Requirement document.*

0.3 THE SOFTWARE REQUIREMENTS DOCUMENT

#1 *The function of a Requirement document is to serve as the mandate or terms of reference for the design, development and realisation of the technical component of a system or subsystem. Usually there is only one Requirement document for each IDA Project, even when the required development encompasses more than one development contract.*

#2 *A User Requirement document is produced as a result of appropriate Requirements Analysis activity, based on the stipulations of the Project Definition*[1] *document and the Global Implementation Plan. All specific requirements in the User Requirement document must be consistent with similar statements in higher-level specifications, if they exist.*

#3 *The User Requirement document is the primary input to subsequent System Design work and to the procurement specifications for pertinent system development contracts.*

#4 *Requirements Analysis will generally include*

- *review of all relevant documents,*
- *establishment of an initial system concept, if not already done*[2]
- *consultation of users*
- *consolidation of inputs*
- *negotiation of a definitive statement of User Requirements.*

#5 *The "user" in the title of this document is a deliberately vague term intended to encompass all parties who have a legitimate interest in the system envisaged, including even customers who will never see it. They are thus sometimes called "stakeholders". They may include, for example*

[2] Requirements are negotiated in the light of the expected feasibility (including cost) of satisfying them. Hence a preliminary/provisional "system concept" is needed before consultation of users.

- *those who will actually operate the telematics system envisaged*
- *those who will operate or manage the processes which the telematics system will support*
- *the external "customers" of those processes*
- *those who will support the system or its users, or provide the platforms on which it runs.*

#6 It must be recognised, however, that users are generally not directly consulted in the course of Requirements Analysis, but only through designated representatives. Some representatives are more closely linked to the population they represent than others are, of course. Nor are all designated representatives fully enfranchised within the development process. In particular, only a limited number have the power to accept or reject the specifications which are produced.

#7 It is important that the nature of the entire user population, the manner in which they were represented during consultation and negotiation, and the way in which specification of User Requirements was (or will be) approved, be spelt out as completely as possible in the User Requirement document.

#8 There may be a conflict of requirements, where there are multiple stakeholders and users. These must be resolved before this document is finalised.

#9 The final decision on approval of the specification of User Requirements should be with the pertinent Sectoral Committee. Therefore:

- *the User Requirement document should not be circulated as other than an unauthorised draft until that approval has been given;*
- *the User Requirement document should be in all respects presented at a level and in a manner suitable for evaluation and approval by the Sectoral Committee.*

1. Introduction

1.1 Purpose

This document is the definitive specification of the user requirements for SELECT agency employee management system as a course project for CSC490 (Software Engineering) At the Lebanese American University. It is a primary input to the technical development of those facilities, and the primary specification of the criteria against which the acceptability of those facilities will be evaluated after they have been developed.

1.2 Document Conventions

This document is written according to the IEEE standard format with some modifications. The writing properties are set by the instructor as follows: text: times new roman 12, chapter: 14 bold, section: 12 bold, line space: single, margin: default

1.3 Intended Audience and Reading Suggestions

a. all responsible for the management of the development

b. users, user representatives, and other interested parties

c. course instructor Karim Youssef

1.4 Product Scope

The developed product will be given to the mentioned agency to help them digitalize their work.

1.5 Feasibility study

What if the system was not implemented?
Employee management is hard to control and organize. The company will go back to using unprofessional, for-another-purpose software which is WhatsApp that is designed for messaging and not handling employees. Time will be wasted, staff costs will increase and less accuracy and more ambiguity will result.

What are current process problems?
WhatsApp contact lists are bloated by the high number of employee responses. Man y staff manage multiple WhatsApp numbers to reduce ambiguity. Confirmed applicants are written on another piece of paper. Broadcast lists are sent to wrong employees increasing human-made errors. Less trust comes with more staff managing large number of events which makes a serious problem.

How will the proposed system help?
The system will digitalize the process. Users can easily apply without any time conflicts, where admins can clearly choose the employees they want to take an event. The system will calculate employee salaries and provide an automatically generated schedule for each event.

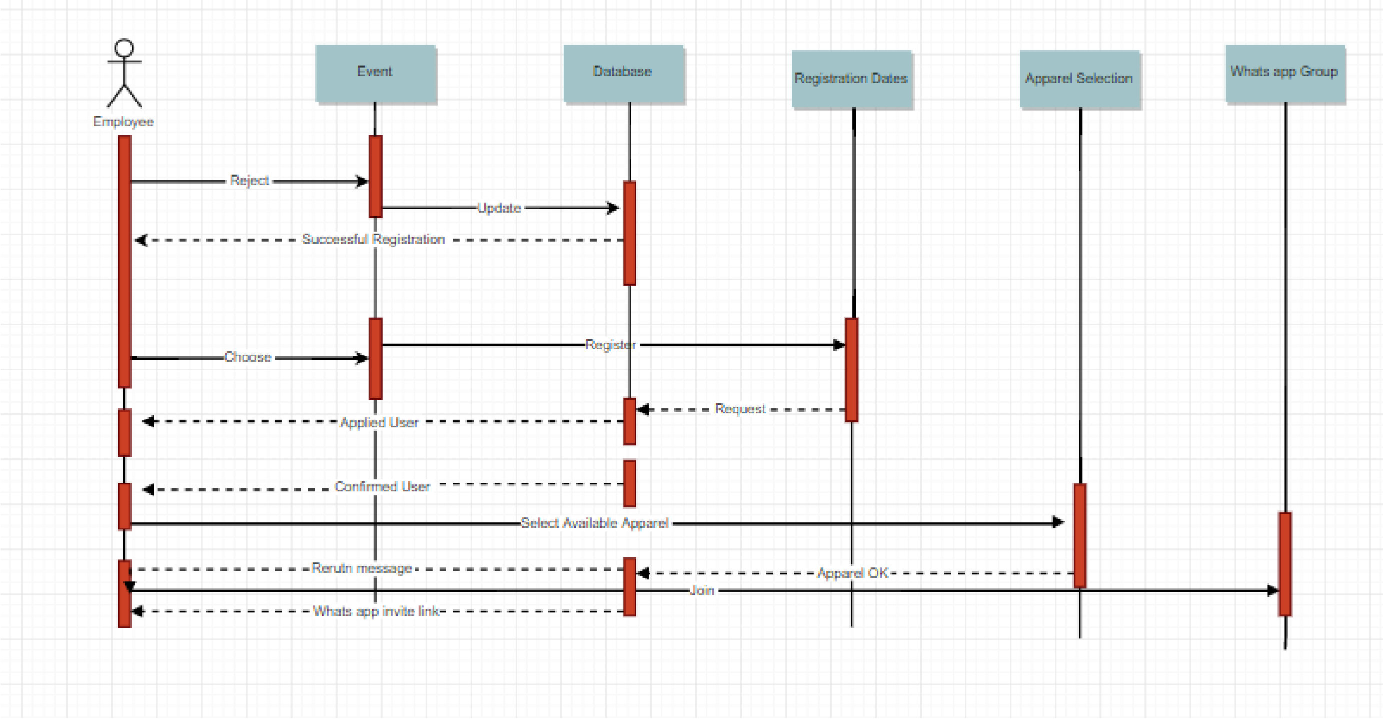
What will be the integration problems?
Credentials must be generated for each employee to join the platform. Some users might find it difficult to install another app on their phone. Staff training to operate the admin panel will be necessary.

Is new technology needed? What skills?
Non-SQL databases must be used. In addition, mobile development should be done using React Native out-of-the-store iOS and android hosting where we can save time and develop a multiplatform application.

1.6 Risk Analysis

- Late Project: remove unimportant and time consuming tasks, and leave only the most important ones.
- Slow server computations: Fallback to an algorithm that may be prone to attacks but does the job.
- Unexpected user changes for the app: We have to make the system maintainable but may lose availability.

1.7 Sequence of Events



2. Glossary

- User: employee
- Admin: manager
- NodeJS: **Node.js**® is a JavaScript runtime built on Chrome's V8 JavaScript engine. **Node.js** uses an event-driven, non-blocking I/O model that makes it lightweight and efficient.
- ExpressJS: Express is a project of the Node.js Foundation. Fork the website on GitHub.
- MongoDB: **MongoDB** is a free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, **MongoDB** uses JSON-like documents with schemas.
- React Native: A **React Native** app is a real mobile app. With **React Native**, you don't build a "mobile web app", an "HTML5 app", or a "hybrid app".
- User Hierarchy diagram: System **hierarchy diagram** is an effective tool in system modeling.
- Data flow diagram: A **data flow diagram** (**DFD**) is a graphical representation of the "**flow**" of **data** through an information system, modelling its process aspects.
- Sequence Diagram: it shows object interactions arranged in time **sequence**.
- Activity network: it is used extensively in project management and is necessary for the identification of a project's critical path (which is used to determine the expected completion time of the project).
- Activity bar chart: it is another way (in addition to the **activity** network **diagram**) of visually rendering the dependencies (or lack thereof) between the various project tasks. This type of **chart** is often referred to as a Gantt **chart**.
- Functional Requirement: it defines a **function** of a system or its component.
- Non-functional requirement: it is a **requirement** that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. It is contrasted with **functional requirements** that define specific behavior or functions.

3. User Requirements Definition

3.1 Functional Requirements

- ✓ the admin shall be able to add events according to specific terms that will be mentioned later
- ✓ the admin shall be able to add employees from an excel sheet to the database
- ✓ the admin shall be able to add another admins
- ✓ the admin shall be able to see who confirmed, rejected and didn't respond to event posts
- ✓ the admin shall be able to see all list of events
- ✓ the admin shall be able to manage events
- ✓ the admin shall be able to register apparels
- ✓ the admin shall be able to send WhatsApp Group invitations
- ✓ the system shall perform in a fast and secure way
- ✓ the system shall use the database in a way that identifiers are unique and redundancy is prevented
- ✓ the employee shall receive WhatsApp invitations and accept them
- ✓ the employee shall be able to receive all posted events
- ✓ the employee shall be able to confirm and reject events

3.2 Non-Functional Requirements

- ✓ Accessibility
- ✓ Usability
- ✓ Efficiency
- ✓ Reliability
- ✓ Portability
- ✓ Delivery:
- ✓ Privacy
- ✓ Safety and security
- ✓ Size
- ✓ Ease of use
- ✓ Fault tolerance
- ✓ Scalability
- ✓ Availability

3.3 Resources

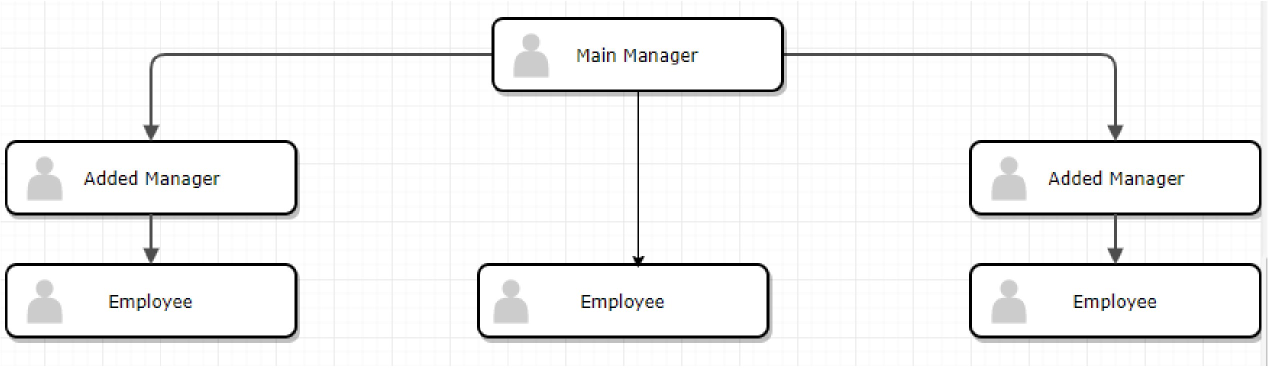
Hardware and software resources:
    No hardware will be used.
Software and services like:
    Vultr, app hosting for android and iOS, website hosting…

3.4 Technologies

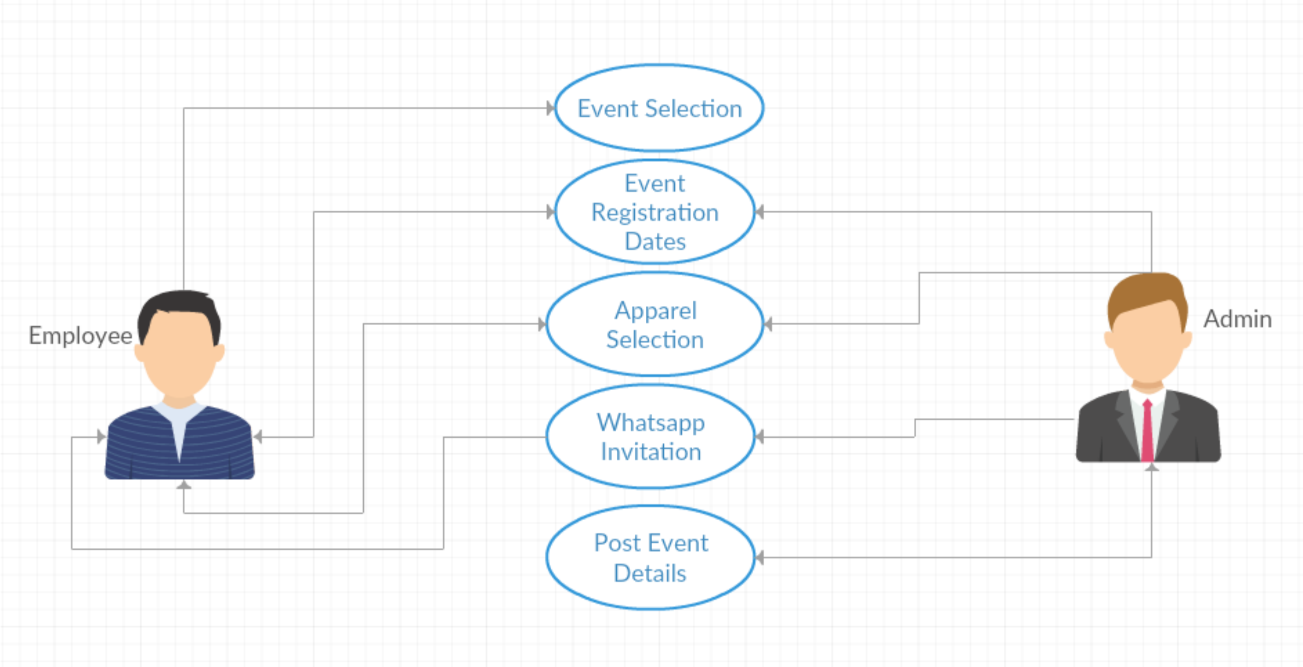
Languages used to develop the system will be NodeJS, ExpressJS, HTML5, CSS3, Bootstrap, JavaScript, MongoDB, React Native…

3.5 User Hierarchy

The system will be given at first to the owner of the agency who can add other admins, giving them the same qualifications. On the other hand, there are the users or the employees in this case who are added by the admins and receive their posts.



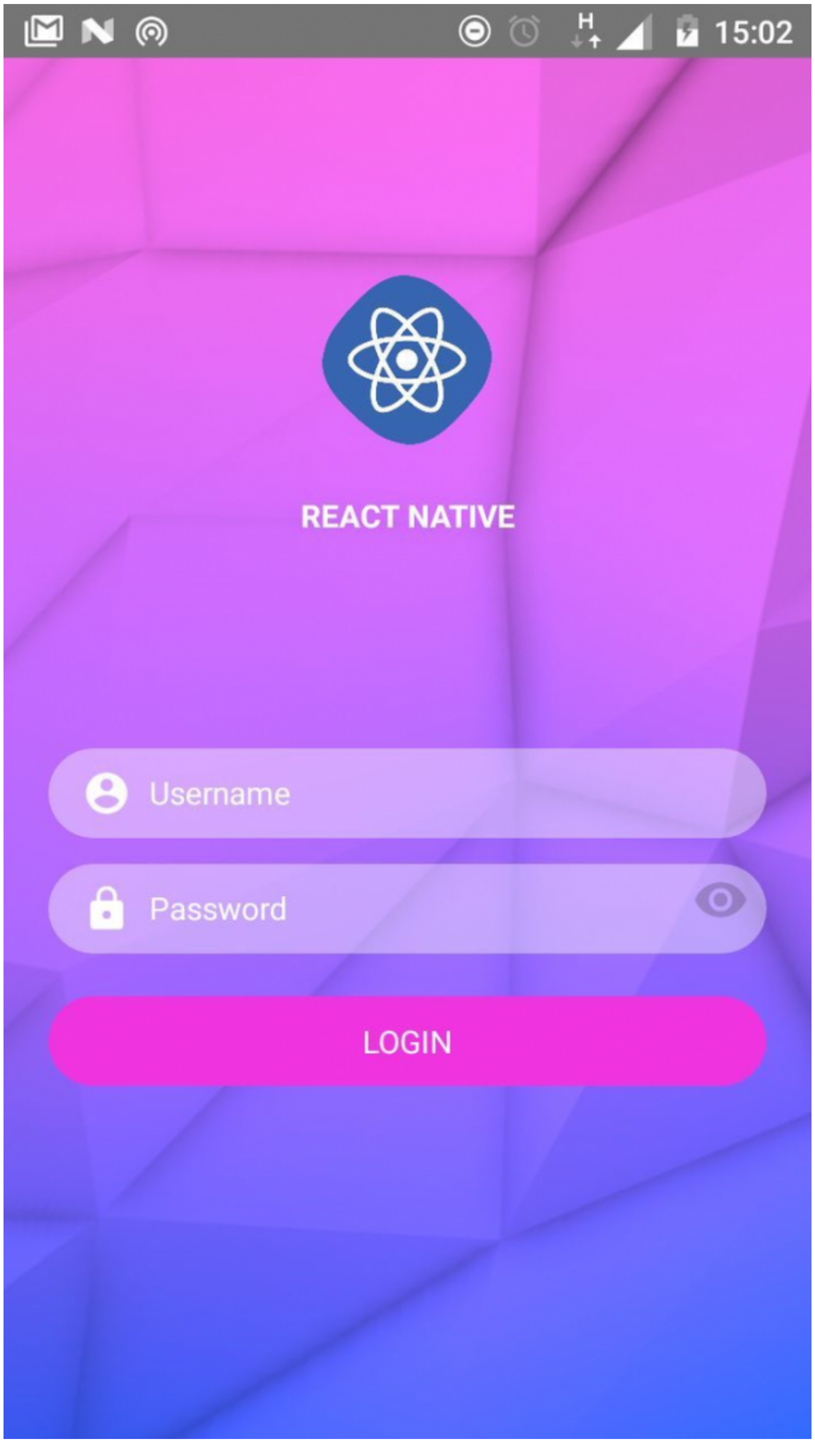
3.6 Use case

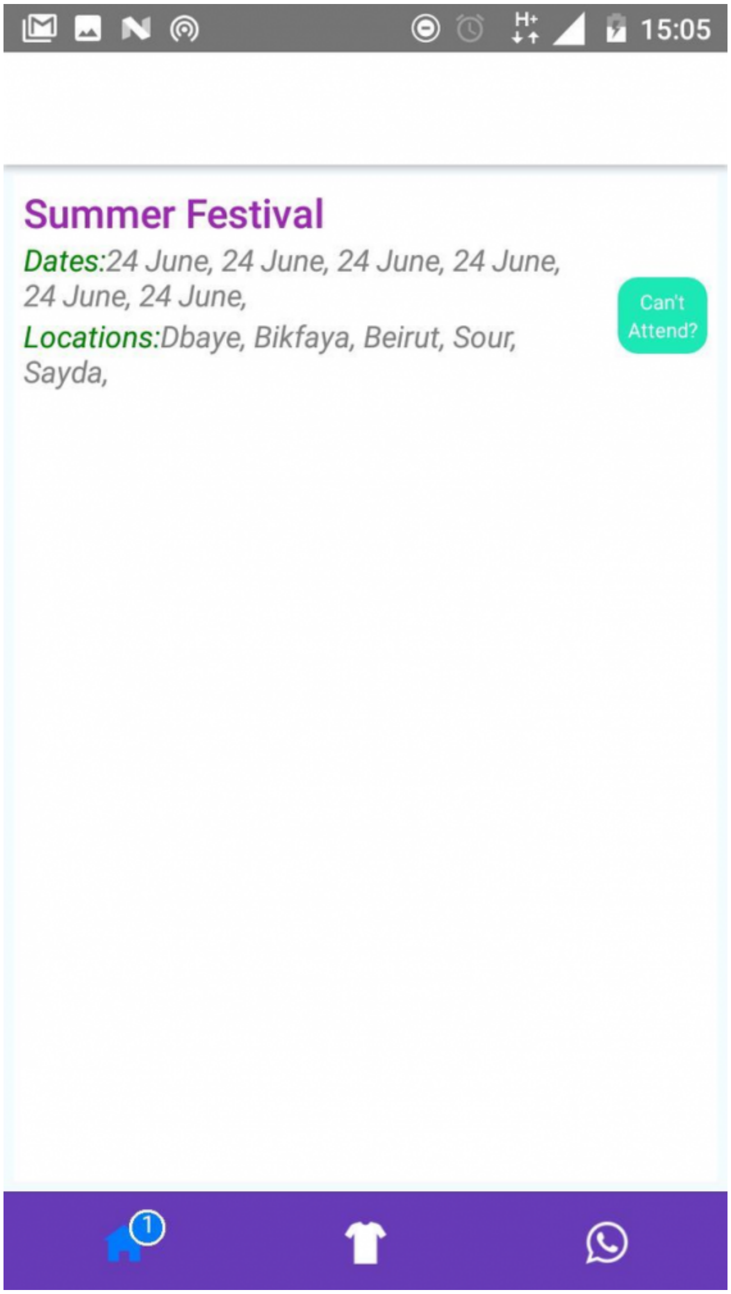


4. System Architecture

4.1 Mobile Application

This platform is designed for employees to check for new events, apply and reject posts. It is the way they communicate with their managers.



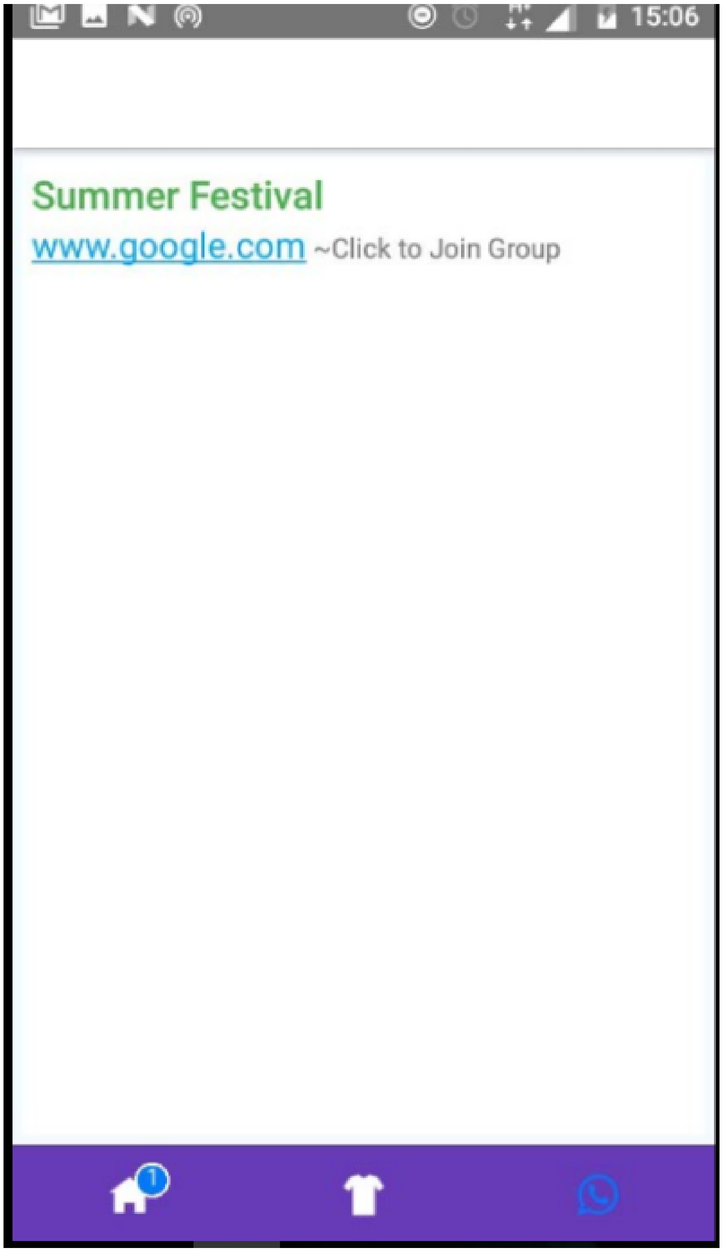


15:05

Summer Festival

*Dates:*24 June, 24 June, 24 June, 24 June, 24 June, 24 June,

*Locations:*Dbaye, Bikfaya, Beirut, Sour, Sayda,



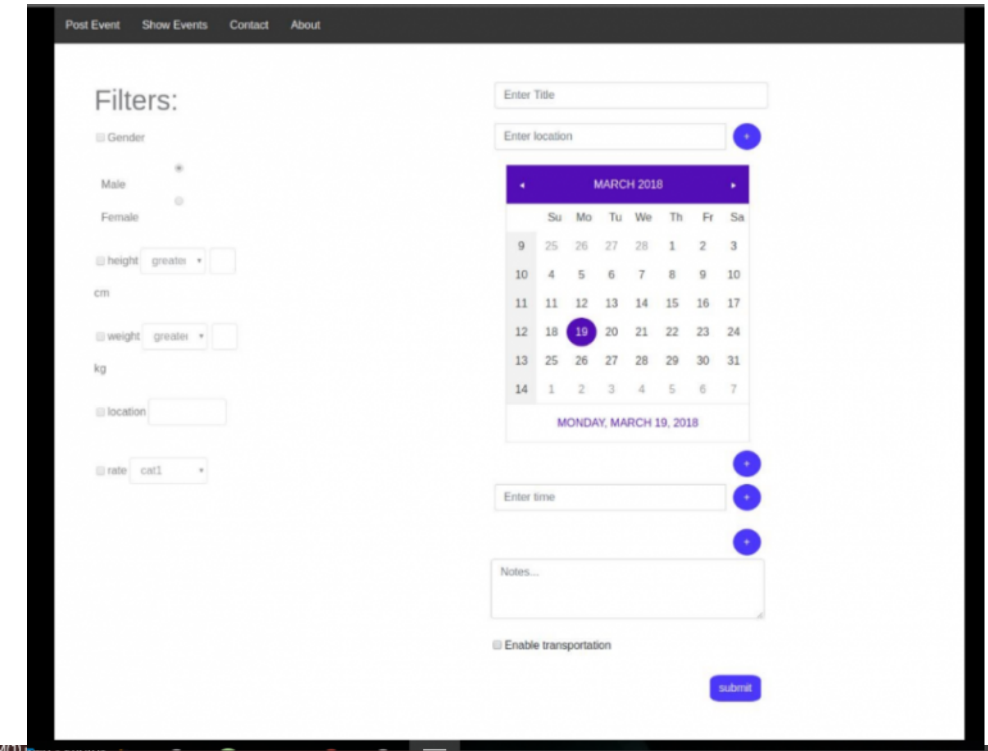
15:06

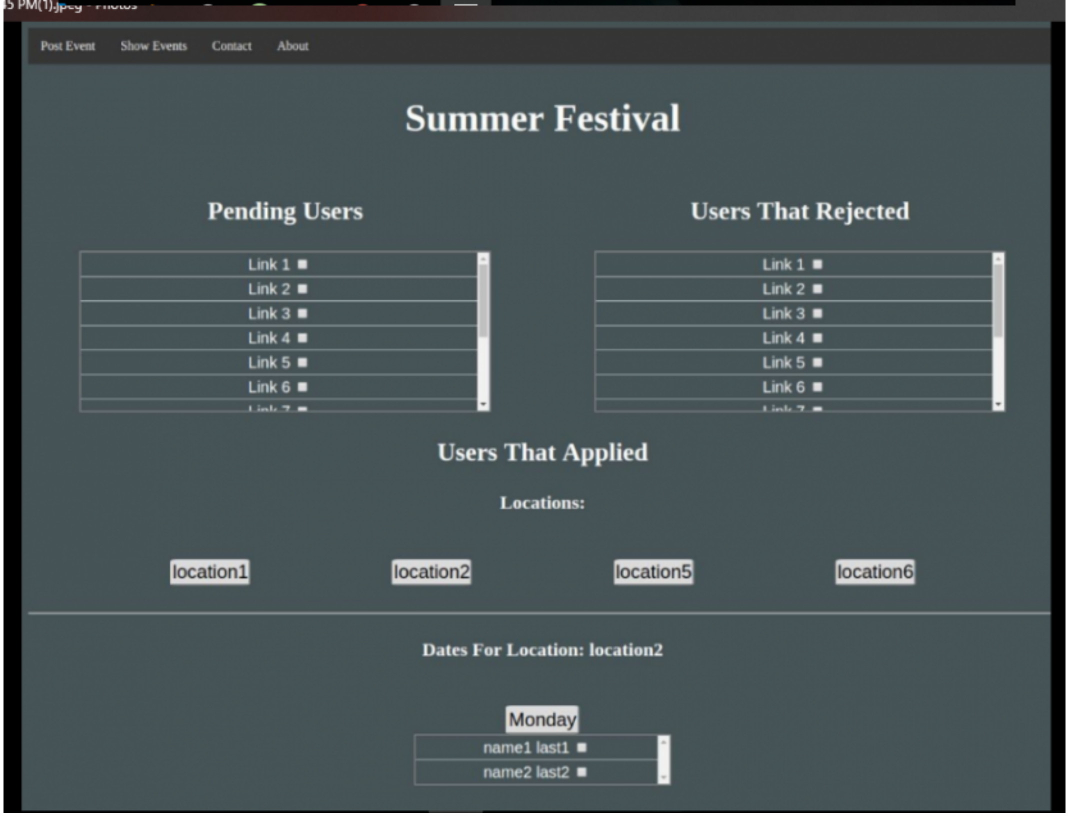
Summer Festival

www.google.com ~Click to Join Group

4.2 Web-based Admin Panel

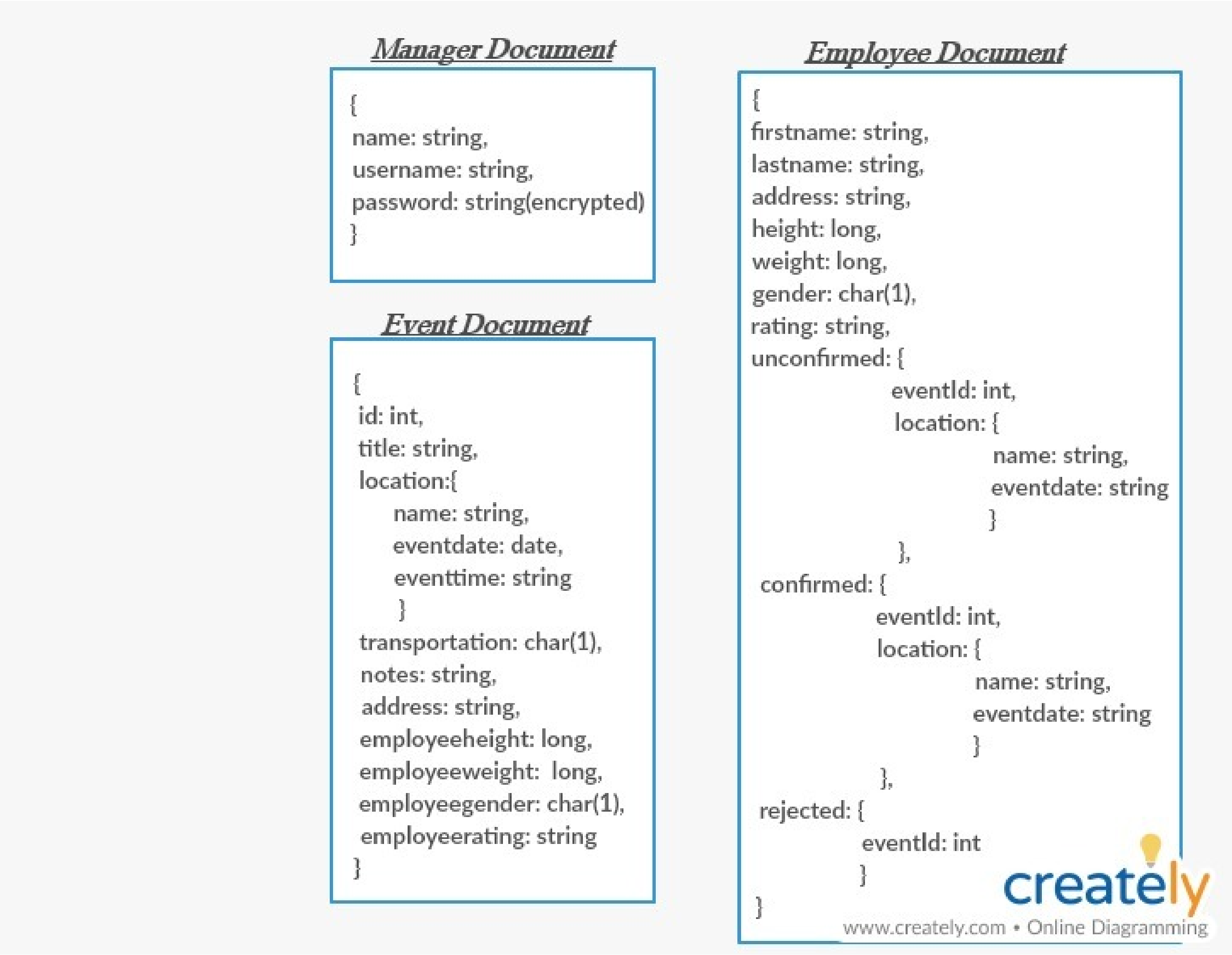
This platform is designed for managers to add events, employees and manage all their functions in an easy and fast way.





4.3 Database:

Which is composed of several documents, each belonging to an object in the system.



Manager Document

```
{
name: string,
username: string,
password: string(encrypted)
}
```

Event Document

```
{
 id: int,
 title: string,
 location:{
      name: string,
      eventdate: date,
      eventtime: string
       }
 transportation: char(1),
 notes: string,
 address: string,
 employeeheight: long,
 employeeweight:  long,
 employeegender: char(1),
 employeerating: string
}
```

Employee Document

```
{
firstname: string,
lastname: string,
address: string,
height: long,
weight: long,
gender: char(1),
rating: string,
unconfirmed: {
                  eventId: int,
                  location: {
                               name: string,
                               eventdate: string
                               }
                  },
 confirmed: {
                eventId: int,
                location: {
                             name: string,
                             eventdate: string
                             }
                },
 rejected: {
              eventId: int
              }
}
```

5. System Requirement Specification

5.1 Functional requirements

- ✓ the admin shall be able to add events
  this will be done by submitting a form composed of the event title, dates and locations… as shown in the next section.
- ✓ the admin shall be able to add employees from an excel sheet to the database
  this feature is done using a JavaScript function that takes the submitted file, traverses it, parses each user's information and submits it to the database one by one
- ✓ the admin shall be able to add another admins
  this is done by adding a username and password which will go to the database and consider this admin qualified for using the panel.
- ✓ the admin shall be able to see who confirmed, rejected and didn't respond to event posts
  separate lists will be shown to the manager as collections of employees with their info so that they can contact them later.
- ✓ the admin shall be able to see all list of events
  this is done to organize things and keep admins up-to-date with all the work done by the agency
- ✓ the admin shall be able to manage events
  this is done by approving employees and sending WhatsApp links…
- ✓ the admin shall be able to register apparels
- ✓ the admin shall be able to send WhatsApp Group invitations
- ✓ the system shall perform in a fast and secure way
  this is the strong advantage of using NodeJS as a backend language where it can perform all the transactions in a fast and manageable way.
- ✓ the system shall use the database in a way that identifiers are unique and redundancy is prevented
  in order to keep the uniqueness of each element and prevent ambiguity.
- ✓ the employee shall receive WhatsApp invitations and accept them
- ✓ the employee shall be able to receive all posted events
  this is done by notifying the user on his/her phone that another event has been posted.
- ✓ the employee shall be able to confirm and reject events
  this is done by a simple user interface with a click of a button where it can either add him/her to the confirmed or rejected collection, otherwise will be marked as unanswered so that the manager can check out all the details as mentioned before.

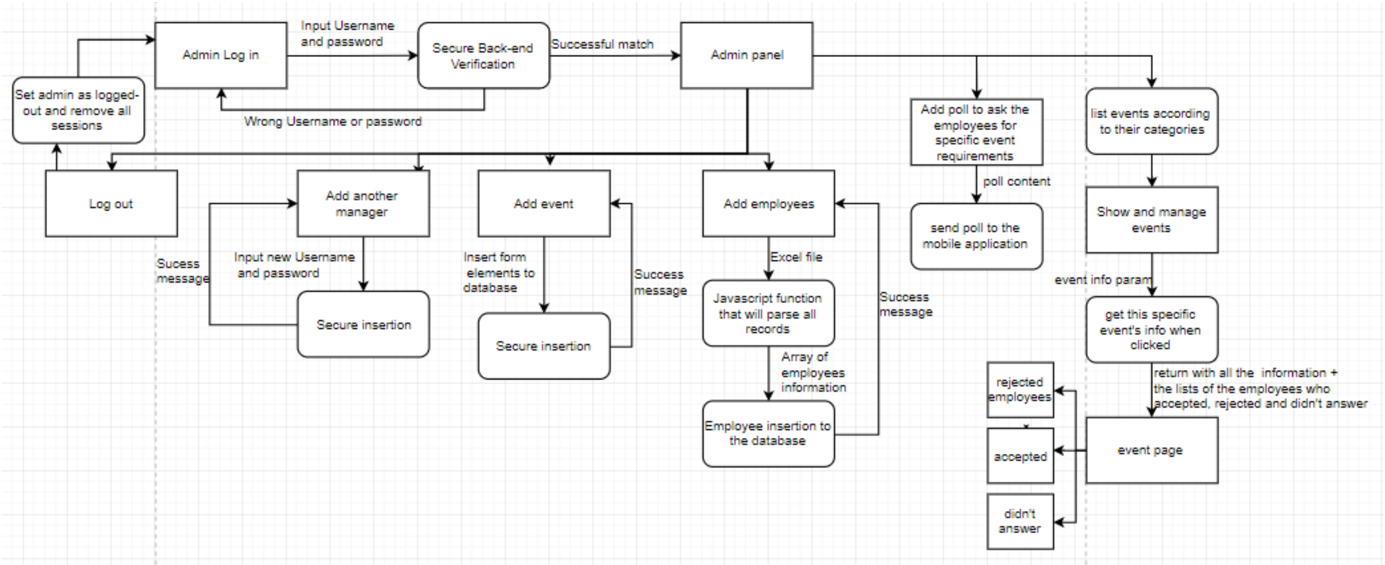
5.2 Non-functional requirements

- ✓ Accessibility: the admin panel should be accessible from the web for managers and the mobile application should be accessed by employees from their phones documentation.
- ✓ Usability: the system should only be offered and used to managers and employees of SELECT agency.
- ✓ Efficiency: in terms of complexity of the used algorithms and the way the transactions take place, both from the performance and space point of view.
- ✓ Reliability: the system should be reliable to help the agency function all of its tasks and daily work normally.

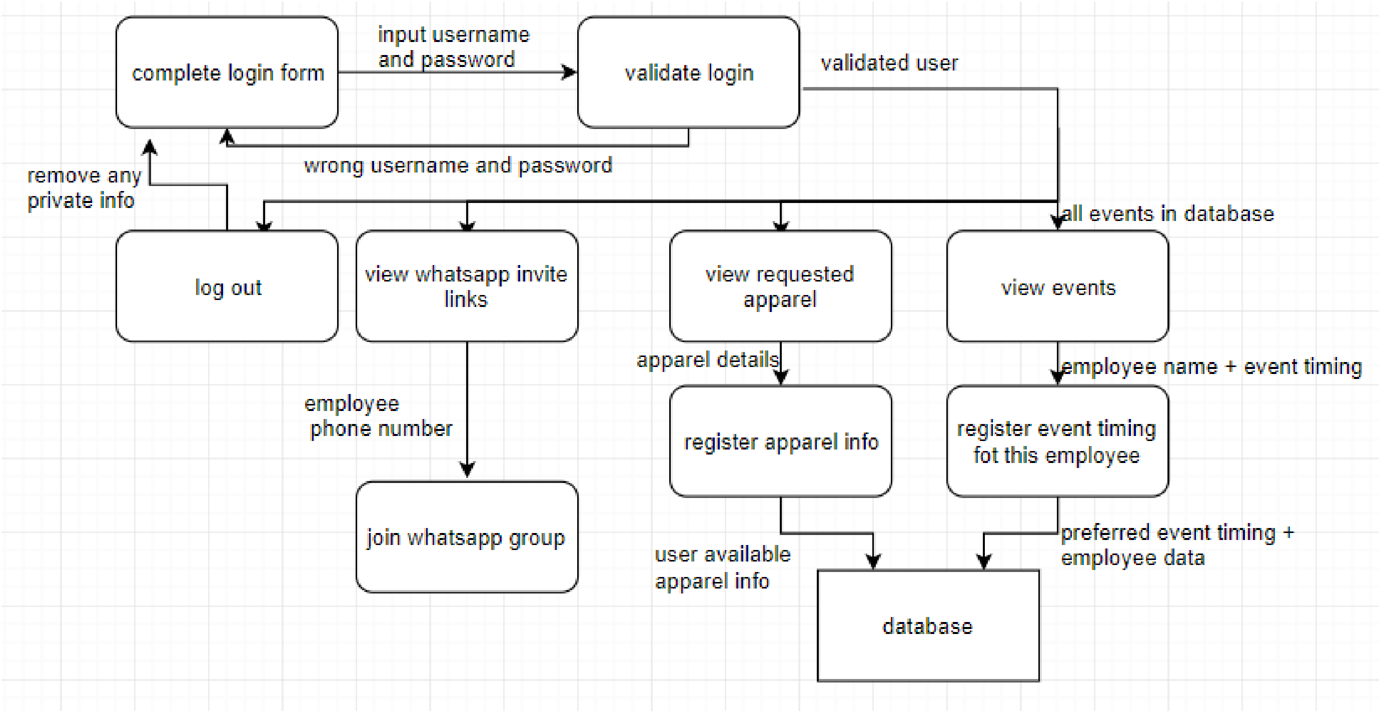
- ✓ Portability: the admin panel will be hosted on the web and the application will be on the phones of the employees.
- ✓ Delivery: events should be delivered to the employees without any errors, and results will be shown on the admin panel.
- ✓ Privacy: all the information will be confidential.
- ✓ Safety and security: the platform will be safe enough to face attacks.
- ✓ Size: the application will be as small as possible in order to be easy to download and use.
- ✓ Ease of use: user interfaces will be simplified as possible.
- ✓ Fault tolerance: the number of errors should be minimized and faults should be prevented.
- ✓ Scalability: the system should be scalable enough to perform all the functions.
- ✓ Availability: the system should be available 24/7 for the customer to use it at any time.

6. System Models

6.1 Web-based Admin Panel Data flow


Admin Log in
Input Username and password
Secure Back-end Verification
Successful match
Admin panel
Wrong Username or password
Set admin as logged-out and remove all sessions
Log out
Add another manager
Sucess message
Input new Username and password
Secure insertion
Add event
Insert form elements to database
Secure insertion
Success message
Add employees
Excel file
Javascript function that will parse all records
Array of employees information
Employee insertion to the database
Success message
Add poll to ask the employees for specific event requirements
poll content
send poll to the mobile application
list events according to their categories
Show and manage events
event info param
get this specific event's info when clicked
return with all the information + the lists of the employees who accepted, rejected and didn't answer
event page
rejected employees
accepted
didn't answer

6.2 Mobile application Data flow



7. System Evolution

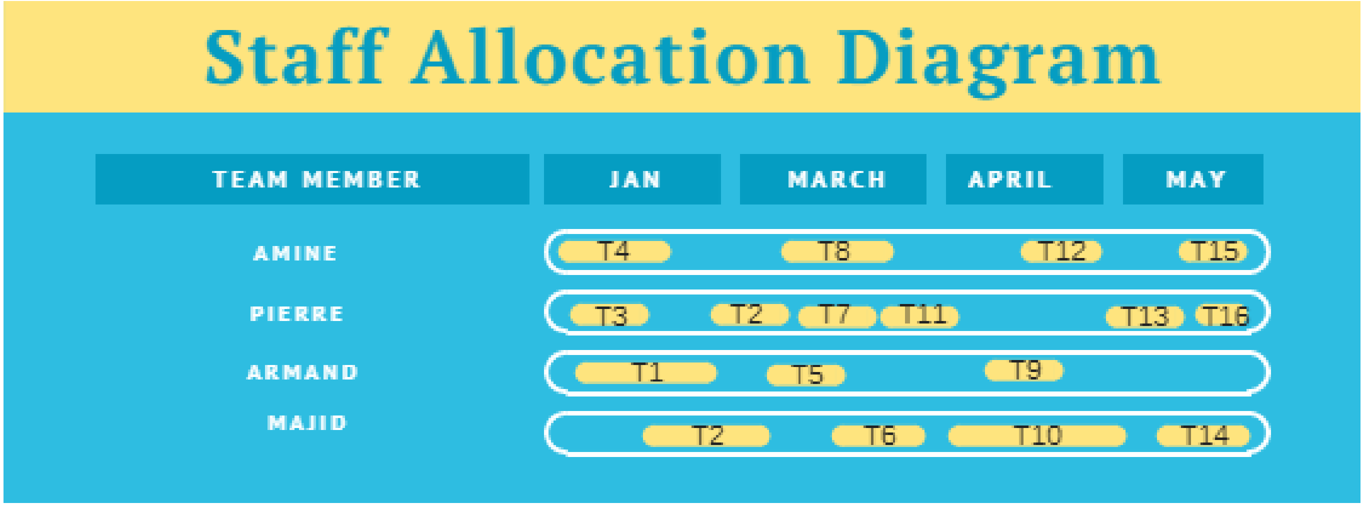
The system was refined once by removing the AngularJS language and replacing it with EmbedJS which will save time and effort.

Other system refinements and updates will take place in case there was a delay or errors for some usage of technologies or features that are not necessary.

Other updates can be determined by the customer.

8. Appendices

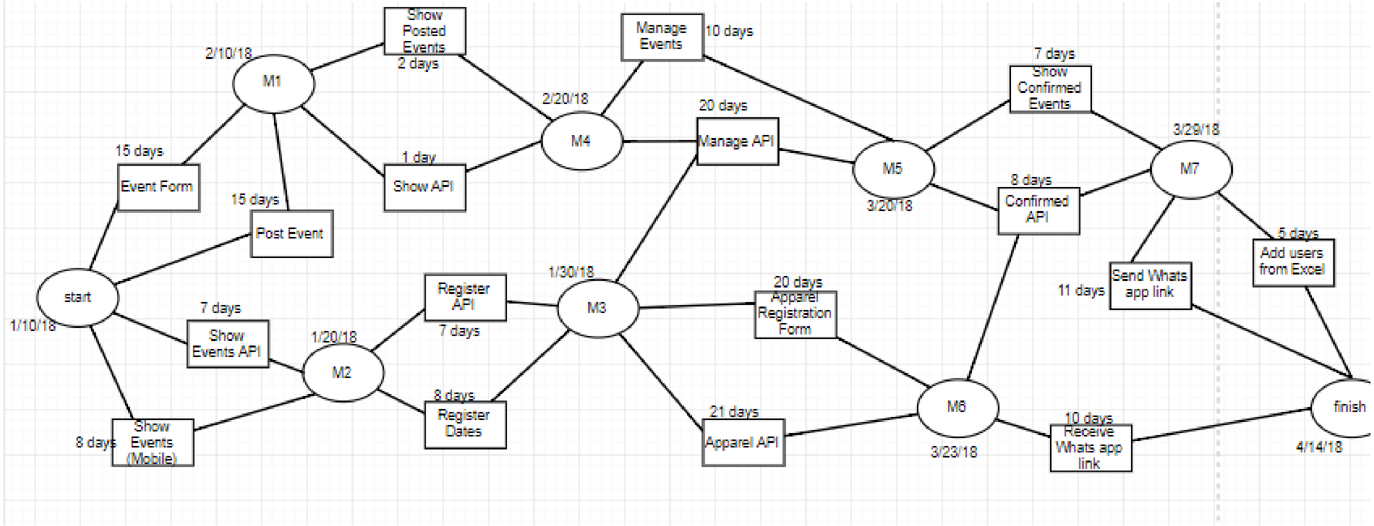
8.1 Staff Allocation



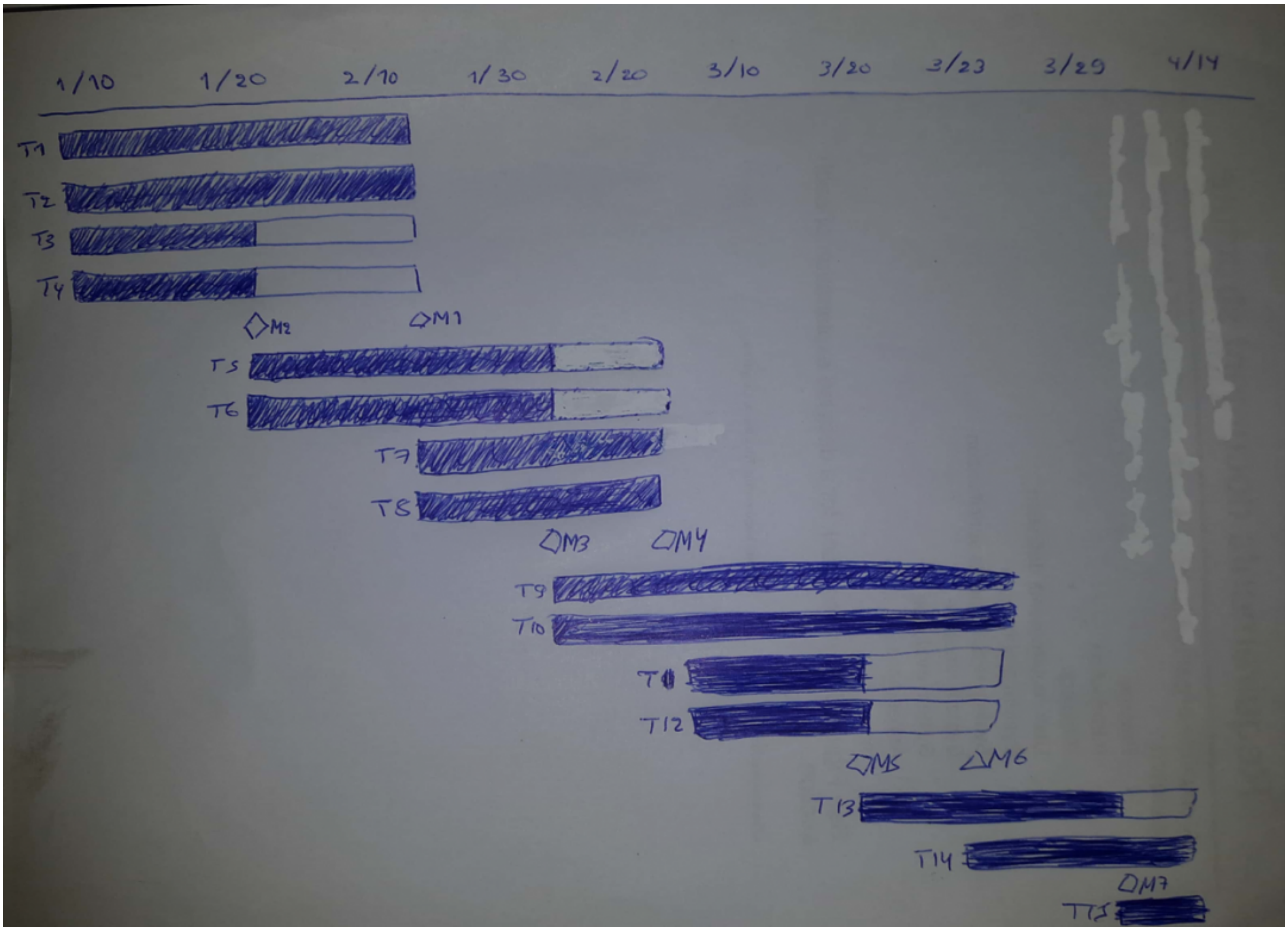
Staff Allocation Diagram

TEAM MEMBER	JAN	MARCH	APRIL	MAY
AMINE	T4	T8	T12	T15
PIERRE	T3	T2 T7 T11		T13 T16
ARMAND	T1	T5	T9	
MAJID	T2	T6	T10	T14

8.2 Activity network



8.3 Activity bar chart


1/10
1/20
2/10
1/30
2/20
3/10
3/20
3/23
3/29
4/14
T1
T2
T3
T4
M2
M1
T5
T6
T7
T8
M3
M4
T9
T10
T11
T12
M5
M6
T13
T14
M7
T15