Technical Design Document

**Nubian**

Prepared by:

Eslam Saad Abd-elmegeed

[Version: 1.0]

[Date: 19/7/2017]

Table of Contents

[1 Introduction 4](#_Toc216541874)

[1.1 Overview 4](#_Toc216541875)

[1.2 Goals and Objectives 4](#_Toc216541876)

[1.3 Definitions 4](#_Toc216541878)

[2 General Design Constraints 5](#_Toc216541881)

[2.1 Product Environment 5](#_Toc216541882)

[2.2 Mandated Constraints 5](#_Toc216541884)

[2.3 Potential System Evolution 5](#_Toc216541885)

[3 Nonfunctional Requirements 5](#_Toc216541886)

[3.1 Usability Requirements 5](#_Toc216541887)

[3.2 Performance Requirements 5](#_Toc216541889)

[3.3 Security Requirements 5](#_Toc216541890)

[3.4 Safety Requirements 5](#_Toc216541891)

[3.5 Legal Requirements 5](#_Toc216541892)

[3.6 User Interfaces 6](#_Toc216541896)

[4 System Features 8](#_Toc216541898)

[4.1 Feature: Calendar view 8](#_Toc216541899)

[4.1.1 Description and Priority 8](#_Toc216541900)

[4.1.2 Use Case: view as a calendar 8](#_Toc216541901)

[4.2 Feature: New feeds 8](#_Toc216541903)

[4.2.1 Description and Priority 8](#_Toc216541904)

[4.2.2 Use Case: following events 9](#_Toc216541905)

[4.3 Feature: Handle events 9](#_Toc216541903)

[4.3.1 Description and Priority 9](#_Toc216541904)

[4.3.2 Use Case: treat with system 9](#_Toc216541905)

[4.4 Feature: Login form 9](#_Toc216541903)

[4.4.1 Description and Priority 9](#_Toc216541904)

[4.4.2 Use Case: sign-up 9](#_Toc216541905)

[4.5 Feature: Favorite list 9](#_Toc216541903)

[4.5.1 Description and Priority 9](#_Toc216541904)

[4.5.2 Use Case: interests 10](#_Toc216541905)

[4.6 Feature: tips for help 10](#_Toc216541903)

[4.6.1 Description and Priority 10](#_Toc216541904)

[4.6.2 Use Case: need for help 10](#_Toc216541905)

[4.7 Feature: Search label 10](#_Toc216541903)

[4.7.1 Description and Priority 10](#_Toc216541904)

[4.7.2 Use Case: finding associations 10](#_Toc216541905)

[4.8 Feature: Location ‘map’ 10](#_Toc216541903)

[4.8.1 Description and Priority 10](#_Toc216541904)

[4.8.2 Use Case: location 11](#_Toc216541905)

[4.9 Feature: Chatting groups 11](#_Toc216541903)

[4.9.1 Description and Priority 11](#_Toc216541904)

[4.9.2 Use Case: chatting 11](#_Toc216541905)

# Introduction

## Overview

This document defines the requirement for the “Nubian” application that is being developed for “Mr. Hossam Samir”. The purpose of this document is to represent the application requirements in a readable way so that client and stakeholders can understand them and verify them for correctness and give enough details about the application to the developers and designers to implement and deliver the software.

This document doesn’t determine *project* issues such as schedule, cost, development methods, deliverables and testing procedures.

The “Nubian” application is an android application for Nubian people. It provides a way for them to keep in touch with each other, and know what is happening and what will happen for a while.

## Goals and Objectives

It's a special application for Nubian people which displays all the current and upcoming events such weddings, condolences, funerals or any other event.

Making the goals and objectives of the product explicit guides developers and give direction to the project. During the development of even a small system. developers make hundreds of tiny decisions (consciously and unconsciously) not specifically addressed by the formal requirements. Knowing the goals and objectives will help them make the best decisions for the product.

Goals and objectives also help keep the project on track. Without a clear set of goals and objectives the scope of the project can grow or shift as new information arrives. Shifting the focus of the project isn’t bad as long as it is done open and explicitly with everyone in agreement.

The three main goals of the “Nubian” application are:

1. Provide the normal users by a set of events occurring in a while.
2. Give the users who will login “Associations” the permission to add, edit or delete events
3. Let the association announce their events in an organized way.

## Definitions

**Product** – what is being described here; the application specified in this document.

**Shall** – adverb used to indicate importance; indicates the requirement is mandatory. “Must” and “will” give the same meaning of “shall”.

**Should** – adverb used to indicate importance; indicates the requirement is desired but not mandatory.

**May** – adverb used to indicate an option. For example, “The system may be taken offline for up to one hour every evening for maintenance.” Not used to express a requirement, but rather to specifically allow an option.

# General Design Constraints

## Product Environment

The product will be in a specific environment and a large certain group of people will use it.

## Mandated Constraints

The client specified that the application must be an android application only using Java.

## Potential System Evolution

* The system may have a chatting groups option in the future.
* The system may have other languages (English or French).

# Nonfunctional Requirements

## Usability Requirements

One method of specifying usability requirements is to specify efficiency and satisfaction for the users by rating the app.

A simpler alternative is to design a survey to measure user satisfaction and get consensus on who will take the survey and what will be considered an acceptable aggregate score.

## Performance Requirements

System startup time should be less than 3 seconds. With 30 concurrent users, no operation should take more than 5 seconds and 95% of the operations should take less than 2 seconds.

## Security Requirements

* Access to data and features may be limited to specific users “associations”.
* We shall keep an audit trail of system use.

## Safety Requirements

There’s no requirements about the safety of the environment. The application doesn’t affect the safety at all.

## Legal Requirements

The ability of adding, editing or deleting event won’t be visible to normal user (who doesn’t login).

## User Interfaces

##### Start screen Favourite selection (events)

##### 

##### Favourite selection (associations) Remove event screen

##### 

##### Login form Add event screen

##### 

##### New feeds (normal users). New feeds (associations)

##### 

##### Favourite list Edit event screen

##### 

# System Features

## Feature: Calendar view

### Description and Priority

The client wants the application to view as a calendar and shows the events of each day separately.

Cost: medium

Risk: low

Value: high

### Use Case: view as a calendar

##### When the user opens the application, the events must be displayed in a calendar.

##### When the user searches for a particular association, its events viewed in a calendar.

## Feature: New feeds

### Description and Priority

##### When an association creates an event, the event must be added in a new feeds screen, to make it for the user to find new events.

Cost: high

Risk: medium

Value: high

### Use Case: following events

##### When the user opens the application, he must see what is new in the new feeds screen as a HOME page.

##### When the user reloads the page, he must see if an association added an event.

## Feature: handle events

### Description and Priority

Give the associations (users who will login) the permission to add, edit, or remove events.

Cost: low

Risk: high

Value: high

### Use Case: treat with system

##### The user just follows the feeds without participating in handling the event.

##### Only the logged in users who has that permission to handle events.

## Feature: Login form

### Description and Priority

We shall make a login form for a particular users (Associations) who will have the permission to handle events.

Cost: low

Risk: low

Value: high

### Use Case: sign-up

##### Only the associations will have access information to the application.

##### There’s no sign-up form even for association, the admin will give each association its information to login.

##### The normal users (who won’t login) will just follow the events.

## Feature: Favorite list

### Description and Priority

The client needs a favorite list to make the user selects his interests and follows their feeds.

Cost: medium

Risk: low

Value: medium

### Use Case: interests

##### When the user opens the application for the first time, he will select all the events and associations he is interested in.

##### After a while, if he wants to change some of his interests, he can update and edit this list whenever he wants.

##### When the selects his interests, they must show first in his feeds.

## Feature: tips for help

### Description and Priority

The client wants to show up tips for help the user in understanding and using the application.

Cost: low

Risk: low

Value: low

### Use Case: need for help

##### When the user opens the application for the first time, the tips must be shown up to help him using the app.

##### The user has the ability to skip these tips if he wants to.

## Feature: search label

### Description and Priority

The user can search for a specific association which he is interested in.

Cost: medium

Risk: low

Value: low

### Use Case: finding associations

##### The user can enter association name in the search label and find it immediately.

## Feature: Location ‘map’

### Description and Priority

The association determines the location of the events to make it easy for normal users to reach the place.

Cost: medium

Risk: medium

Value: high

### Use Case: location

##### When the user opens an event, he would be able to know the location of the event on map by clicking on the map.

## Feature: chatting groups

### Description and Priority

The association has the ability to make a chat group to keep in touch with the users.

Cost: high

Risk: high

Value: medium

### Use Case: chatting

##### When the user enters an association, he can join the chat group by clicking button “join chat” and the association accepts the request.