

MeNoon

**Project NameTeam NameDocument Namex.yCM\_IdMmm dd, yyyy**

**<Project Name>****Statement of Work****<x.y>****<CM Identifier>****<Mmm dd, yyyy>**

Programming camps website

Software Requirements Specification (SRS)

Team nameProject nameTeam nameVersionCm IdDateVersion: 1.0

G2\_SE01\_v1.0

Revision History

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sl. No. | Prepared/  Modified by | E-mail | Version | Date | Approved by | Descriptions/  Remarks |
| 1. | 1-Hager Hosny  2-Reem Omar  3-Reem Amr | Hagarhosny19@gmail.com | 1.0 | 3/15/2017 |  |  |

Distribution list

|  |  |  |
| --- | --- | --- |
| **Name** | **E-mail** | **Notes** |
| 1-Hager Hosny  2-Reem Amr  3-Reem Omar | Hagarhosny19@gmail.com |  |

Table of Contents

1. Introduction 5

1.1 Purpose 5

1.2 Scope 5

1.3 Definitions, Acronyms and Abbreviations 5

1.4 References 5

1.5 Overview 5

2. Market Survey 5

2.1 <Project Name> 5

**2.1.1** **Project Description** 5

**2.1.2** **Functional Specifications** 6

**2.1.3** **Non-Functional Specifications** 6

**2.1.4** **Limitations** 6

2.2 <Project Name> 6

2.3 <Project Name> 6

2.4 Summary 6

**2.4.1** **Functional Specifications** 6

**2.4.2** **Non-Functional Specifications** 6

**2.4.3** **Limitations** 7

3. Cost Estimate 7

4. Feasibility Study 7

5. Requirements Specifications 7

5.1 User Requirements (User Stories) 7

**5.1.1** <Requirement Description and priority> 8

**5.1.2** <Requirement Description and priority> 8

5.2 System Requirements (Use Cases) 8

**5.2.1** <Use Case Description> 8

**5.2.2** <Use Case Description> 8

**5.2.3** <Use Case Description> 8

5.3 Non-functional Requirements 8

**5.3.1** <Requirement Description and the user stories ID that this requirement is addressing> 9

**5.3.2** <Requirement Description> 9

6. High level plan 9

7. Supporting Information 9

# Introduction

The purpose of this document is to present a detailed description of the programming camps website. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate.

The system will allow the users to apply for a certain course/event and pay the fees.

## Purpose

The system is a programming camps online reservation website for a company called MeNoon LLC. MeNooN LLC specializes in software development and advanced computer science training courses. The company encourages young people to take competitive programming courses to boost their creativity and strengthen their candidacy for employment abroad.

The system will allow the users to apply for a certain course/event. It should not allow the user to apply if he/she does not have an approved account. If the user has an approved account, he will pass through an IQ test. In case he gets an acceptable score, he will be able to pay the fees and enroll in the event he has chosen.

There are three types of users to our system.

First user: (guest)

The system will allow him to browse the content of the courses/events but he can’t apply.

He/she should be able to contact the company through a form, email or phone, See the feedback of the previous students and browse the videos and pictures of the different events.

Second user: (registered)

The system shall store his information in database .Registered users have to upload, birth certificates, national ID, passports, all these information are optional, however,

they can’t get full access to any of the services unless one of them is uploaded and verified by an admin.

Users can tag themselves ONLY themselves in photos and videos in the “Media”.

Third user: (administrators)

The system should be able to create several types of them and specify their view.

Each admin has a different role

Full access - Interviewer for a special event - Approval - Event creator - Media Uploader –

Tagger - Special Event Attendance - Data Entry.

## Scope

<A brief description of the software application that the SRS applies to; the feature or other subsystem grouping; what Use Case model(s) it is associated with, and anything else that is affected or influenced by this document.>

## Definitions, Acronyms and Abbreviations

<This subsection should provide the definitions of all terms, acronyms, and abbreviations required to interpret properly the SRS.  This information may be provided by reference to the project Glossary.>

## References

[This subsection should provide a complete list of all documents referenced elsewhere in the SRS. Each document should be identified by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.]

## Overview

The document describes the main user and system functionalities. It describes different use cases of the system. In addition to this, it describes the constraints imposed on the system.

# Market Survey

## Switch up website

### **Project Description**

Switch up website is a website offering variety of coding and design boot camps. It allows you to view the curriculum of the courses, their location, number of hours so you can apply for these courses.

### **Functional Specifications**

**<**This should provide a structured listing (1, 1.1, 1.2, etc.) of the functional specification in this project. It is better to group the specification into categories. This will help you to classify specifications in the summary section>

### **Non-Functional Specifications**

**<**This should provide a structured listing (1, 1.1, 1.2, etc.) of the non-functional specification in this project. It is better to group the specification into categories. This will help you to classify specifications in the summary section>

### **Limitations**

**<**You should look for limitations present in this project and their reasons>

## <Project Name>

## <Project Name>

## Summary

<This should summarize the previous section. You should classify the specifications above into categories and specify the essential and non-essential specifications>

### **Functional Specifications**

**<**This should provide a structured listing of the functional specification in similar projects grouped into categories. Write two sections for essential and non-essential specifications>

#### ***Essential Specifications***

##### Registration

* In order to apply for course or event, user should be registered member in our website.
* Admin should approve user account
* Admin should add other admin account

##### Tagging in media

* Tagger Admin should tag users who applied for an event in photos, uploaded in media section
* User should tag himself in photos appearing in media section.
* User should view tagged photo in my media page
* Tagger Admin should approve user tag request in photo

##### Viewing others feedback

* User should view other feedback of students or other users

##### Interviews

* interviewer Admin should accept or refuse interviewees. and give them reason upon accepting or refusing.
* interviewer Admin should view list of interviewees who applied for events.
* User should request interview and choose free slot

##### Courses and Events

* User should browse list of courses.
* User should view upcoming events.
* User should apply for a upcoming events.

##### About section

* Displaying information about system.

##### Home section

* User should customize home in order to see events/news according to his/her country or according to date

##### Media section

* User should view photos and videos of events

##### Event rules

* User Should view rules of events
* Admin Should put/assign rules for event

#### ***Non-Essential Specifications***

##### Subscribing to news letter

* User shall sign up for newsletter

##### Give feedback

* User shall send feedback to organization

##### Contact organization

* User should contact company through phone, email, or form.

### **Non-Functional Specifications**

**<** This should provide a structured listing of the non-functional specification in similar projects grouped into categories. Write two sections for essential and non-essential specifications>

#### ***Essential Specifications***

##### Security

* Since finger prints, passport number, national id is stored in the database there must be good security to protect them.
* This can be solved by using Public-key cryptography, or asymmetric cryptography.

##### Efficiency

* Avoid landing page redirects.
* Enable compression.
* improve server response time

##### Usability

* The website should be user friendly, easy to use and follows specific UI design standards.
* Configure the viewport
* Size content to viewport
* Size tap targets appropriately.
* Use legible font sizes.

.

#### ***Non-Essential Specifications***

### **Limitations**

**<**You should summarize the limitations present in these projects and analyze their reasons. Discuss how your project may overcome these limitations>

# Cost Estimate

|  |  |  |
| --- | --- | --- |
| Item | Estimated Cost | |
| US Dollars | L.E. |
| 1. **Development Cost** | 300 | 6000 |
| 1.1. Designers | 100$ | 1800 |
| 1.2. Developers | 150$ | 2700 |
| 1.3. Tester | 50$ | 900 |
| 1.4. Programmers | 100$ | 1800 |
| Total | 400$ | 7200 |
| **2. Deployment Cost** |  |  |
| 1.1 Host | 9$ | 162 |
| 1.2 Domain | 16$ | 288 |
| Total | 25$ | 450 |
| Total Estimated Cost | 425$ | 7650 LE |

**Table 2: Cost Estimate**

# Feasibility Study

## Organizational objectives:

1. It will add full functional website that serves the organization in managing its Camps.
2. System will provide more security and maintainability to its Camps, events media

## Technical

1. System will be developed using Laravel framework technology
2. Project will be done within budget as expenditures are all fixed and not price changing
3. Time constraints (e.g. exams, other projects, deadlines, ...etc) will be handled effectively by tasks assignment on well managed time plan

## System Integration

# Requirements Specifications

## User Requirements (User Stories)

<The services provided for the user should be described in this section. This section should be understandable by system users who do not have detailed technical knowledge. Each requirement should take a unique identifier as follows. Remember that you will be using these identifiers later on in the design, implementation and even testing phases>

<This should give a brief description for each functional requirement along with the importance priority for each according to the user>

### Creating account for admins

ID: US1

Description: As a full functional admin, I want to add accounts by entering by adding Full name, email, password,

Expiry date, activation date for newly recruited admins

### Accepting or refusing interviewees

ID: US2

Description: AS an Event interviewer, I want to accept or refuses interviewed user to the event ang giving them reasons for accepting or rejection.

### Uploading videos and photos

ID: US3

Description: As a media admin, I want to upload photos and videos to website by navigating to media page and pressing add new photo or videos button.

### Adding finger prints

ID: US4

Description: As a data entry admin, I want to add finger prints to existing user of the system who attended events and didn’t upload their own fingerprints

### Deleting Account

ID: US5

Description: As a full functional admin, I want to delete or deactivate other admins account.

### Tagging users

ID: US6

Description: As a Tagger admin I want to tag users in the uploaded photos in the media page.

## System Requirements (Use Cases)

<The services provided for the user should be described in this section from the system view not a user view. This section would be understandable by the development team. Each requirement should take a unique identifier as follows and map to the user story. Remember that you will be using these identifiers later on in the design, implementation and even testing phases.>



### Sign Up use case

**Name: Sign up**

**Identifier** UC1

**Preconditions**

1. The user has opened the website and the home page has appeared.

**Basic Course**

1. Use case begins when the user clicks the sign up tab in the navigation bar.
2. A modal appears and the user enters full name, username, password, email, address.
3. If the information is correct, the user returns to home page and a profile tab appears in the navigation bar.

**Alternate Course A:**

**Condition:** User enters incorrect information.

1.The user is alerted and is prompt to re-enter his information.

**Post conditions** (*List the state(s) the system can be in when this use case ends*)

1. The user returns to home page.
2. A profile tab appears in the navigation bar.

**Map to:** (*List the Identifiers of the user stories that this use case is addressing*)

### View Events use case

**Name: View Events**

**Identifier** UC2

**Preconditions**

1. The user has opened the website and the home page has appeared.

**Basic Course**

Use case begins when the user scrolls down or clicks the event tab in the navigation bar.

**Post conditions**

The user is directed to the events’ media which are displayed in honey comb view.

**Map to:** (*List the Identifiers of the user stories that this use case is addressing*)

### Log in as a normal user use case

**Name: Normal user logs in**

**Identifier** UC3

**Preconditions**

The user goes to the main page of website and can see the log in button

**Basic Course**

1.Use case begins when the user clicks login from the navigation bar.

2.The user is prompt to enter his username and password.

3.Use case ends when the user submits his credentials correctly.

**Alternate Course A: Wrong username or password.**

**Condition:** The user enters wrong username or password

A.1 A message appears telling the user to re-enter his credentials.

A.2 Use case ends when the user submits his credentials correctly.

**Alternate Course B: Forget password**

**Condition:** User clicks the forget password button.

The user goes through the step of getting a new password as in the forget password use case.

**Post conditions**

1.The user is logged in and a “Go to profile” button appears in the navigation bar of the home page.

**Map to:** (*List the Identifiers of the user stories that this use case is addressing*)

### Log in as an admin use case

**Name: Admin logs in**

**Identifier** UC4

**Preconditions**

1.The user goes to the main page of website and can see the log in button

**Basic Course**

1.Use case begins when the user clicks login from the navigation bar.

2.The user is prompt to enter his username and password.

3.Use case ends when the user submits his credentials correctly.

**Alternate Course A: Wrong username or password.**

**Condition:** The user enters wrong username or password

A.1 A message appears telling the user to re-enter his credentials.

A.2 Use case ends when the user submits his credentials correctly.

**Alternate Course B: Forget password**

**Condition:** User clicks the forget password button.

The user goes through the step of getting a new password as in the forget password use case.

**Post conditions**

1.The user goes to the admin profile page.

**Map to:** (*List the Identifiers of the user stories that this use case is addressing*)

### Apply for a course use case

**Name:** Apply for a course

**Identifier** UC5

**Preconditions**

1. A user logs in as in the Log in use case, navigates to courses (upcoming events) and the apply for a course button is activated.

**Basic Course**

1.Use case begins when the user chooses to apply for a certain course.

2.If the user has registered and his information is complete, then the user is prompt to take an IQ test.

3.If the user’s score meets the minimum score required to enroll in the course, user is prompt to pay as in Pay fees use case.

4.If the user didn’t pass the test, he would be asked to try again later.

5.Use case ends when the user either pays for the course or exits due to failure.

**Alternate Course A: User’s information is not complete**

**Condition:** If the user hasn’t submitted the passport photocopy or national id or birth certificate.

A.1 The user is prompt to complete his information

A.2 The user has to wait for admin approval and re-apply for the course.

**Alternate Course B: No IQ test required**

**Condition:** If an IQ test is not required to apply for the course

B.1 The user proceeds directly to payment provided that his information is complete.

**Post conditions**

The admin can see the user as one of the applicants in the course.

The user’s payment status is updated in the database.

**Map to:** (*List the Identifiers of the user stories that this use case is addressing*)

### Tag in media use case

**Name: Tag in media**

**Identifier** UC6

**Preconditions**

1.The user logs in and scrolls down to events in the main page.

**Basic Course**

1.Use case begins when a user clicks a photo to tag himself in.

2. The user clicks tag.

3.A message appears telling the user that his tag will appear when the admin approves it and verifies that he is in the photo.

**Post conditions**

1.The user’s tag request is added to the database, so that the admin can regard its approval.

2.The user expects to get the photo added to his media if his tag request is approved by the admin.

**Map to:** (*List the Identifiers of the user stories that this use case is addressing*)

### Give feedback use case

**Name: Give feedback**

**Identifier** UC7

**Preconditions**

1.The user goes to main page where a feedback button appears.

**Basic Course**

1.Use case begins when the user clicks the “give feedback button”

2.A text box appears where the user can write his own feedback about the website or courses.

3.Use case ends when the user submits his feedback.

**Post conditions**

1.A message appears thanking the user for submitting the feedback.

2.The user’s feedback is added to the database.

**Map to:** (*List the Identifiers of the user stories that this use case is addressing*)

### Review other users’ feedback use case

**Name: Review other users’ feedback**

**Identifier** UC8

**Preconditions**

1. The user goes to the main page where “review feedback” button appears.

**Basic Course**

1.Use case begins when the user clicks the “review feedback” button.

2.A list of previous users’ feedback appears.

3.Use case ends when the user hides the feedback list.

**Post conditions**

1.The user remains at the main page and can continue surfing through the website.

**Map to:** (*List the Identifiers of the user stories that this use case is addressing*)

### Browse courses use case

**Name: Browse courses**

**Identifier** UC9

**Preconditions**

The user goes to the main page where an “events” button appears in the navigation bar.

**Basic Course**

1.Use case begins when the user clicks the “events” button from the navigation bar.

2.The user is taken to the courses page where he can view courses details such as the contents of the course.

3.If the user is logged in, he can see an activated button “apply for a course”. Otherwise, that button is deactivated.

4.Use case ends when the user closes the courses page.

**Alternate Course A: A logged in user wants to apply for a course**

**Condition:** A logged in user clicks the “apply for a course” button

The user goes through the steps indicated in the “apply for course” use case.

**Post conditions**

The user goes back to the main page of the website.

**Map to:** (*List the Identifiers of the user stories that this use case is addressing*)

### Subscribe to newsletter use case

**Name: Subscribe to newsletter**

**Identifier** UC10

**Preconditions**

1.The User is in the main page of the website where the “subscribe to newsletter” option appears.

**Basic Course**

1.Use case begins when the user clicks “subscribe to newsletter”.

2.The user enters his email.

3.Use case ends when the user’s email is submitted and a message appears telling the user that he has successfully subscribed to the newsletter.

**Alternate Course A: User’s email already exists.**

**Condition:** User has subscribed to the newsletter before.

A.1 A message appears telling the user that he has already subscribed before.

**Post conditions**

The user remains at the main page of the website and his email is added to the database if it hasn’t been added before.

**Map to:** (*List the Identifiers of the user stories that this use case is addressing*)

### Verify users’ tags use case.

**Name: Verify users’ tags**

**Identifier** UC11

**Preconditions**

1. Some users have submitted a tag request as in the “tag in media” use case.

2.An admin opens his admin profile page.

**Basic Course**

1.Use case begins when the admin chooses to view tag requests.

2.If the admin is a full access admin or a tagger admin, a list of requests appears and the admin can then

approve or reject users’ tags.

1. Use case ends when the admin hides the tag requests.

**Alternate Course A: The admin is not authorized to deal with tag requests.**

**Condition:** The admin has no full access and is not a tagger admin.

A.1 The admin is not allowed to view the users’ tags.

A.2 A message appears telling the admin that he has no authorization to view the users’ tags.

**Post conditions**

1.Approved user tags are marked as approved in the database.

2.The users whose tags have been approved can now see the media they tagged themselves in in their media page.

3.Users whose tags have been rejected will have their tags deleted from the database.

**Map to:** (*List the Identifiers of the user stories that this use case is addressing*)

### <Use Case Description>

**Name:**

**Identifier** (*A unique identifier for this use case, e.g. UC10*)

**Preconditions** (*List the state(s) the system can be in before this use case starts*)

**Basic Course** (*Describe the “normal” processing path, aka, the Happy Path*)

1. Use case begins when …
2. Use case ends when …

**Alternate Course A: Description of the alternate course**

**Condition:** Indicate what happened

1. List the steps

**Post conditions** (*List the state(s) the system can be in when this use case ends*)

**Map to:** (*List the Identifiers of the user stories that this use case is addressing*)

### <Use Case Description>

**Name:**

**Identifier** (*A unique identifier for this use case, e.g. UC10*)

**Preconditions** (*List the state(s) the system can be in before this use case starts*)

**Basic Course** (*Describe the “normal” processing path, aka, the Happy Path*)

1. Use case begins when …
2. Use case ends when …

**Alternate Course A: Description of the alternate course**

**Condition:** Indicate what happened

1. List the steps

**Post conditions** (*List the state(s) the system can be in when this use case ends*)

**Map to:** (*List the Identifiers of the user stories that this use case is addressing*)

### <Use Case Description>

**Name:**

**Identifier** (*A unique identifier for this use case, e.g. UC10*)

**Preconditions** (*List the state(s) the system can be in before this use case starts*)

**Basic Course** (*Describe the “normal” processing path, aka, the Happy Path*)

1. Use case begins when …
2. Use case ends when …

**Alternate Course A: Description of the alternate course**

**Condition:** Indicate what happened

1. List the steps

**Post conditions** (*List the state(s) the system can be in when this use case ends*)

**Map to:** (*List the Identifiers of the user stories that this use case is addressing*)

### <Use Case Description>

**Name:**

**Identifier** (*A unique identifier for this use case, e.g. UC10*)

**Preconditions** (*List the state(s) the system can be in before this use case starts*)

**Basic Course** (*Describe the “normal” processing path, aka, the Happy Path*)

1. Use case begins when …
2. Use case ends when …

**Alternate Course A: Description of the alternate course**

**Condition:** Indicate what happened

1. List the steps

**Post conditions** (*List the state(s) the system can be in when this use case ends*)

**Map to:** (*List the Identifiers of the user stories that this use case is addressing*)

### <Use Case Description>

**Name:**

**Identifier** (*A unique identifier for this use case, e.g. UC10*)

**Preconditions** (*List the state(s) the system can be in before this use case starts*)

**Basic Course** (*Describe the “normal” processing path, aka, the Happy Path*)

1. Use case begins when …
2. Use case ends when …

**Alternate Course A: Description of the alternate course**

**Condition:** Indicate what happened

1. List the steps

**Post conditions** (*List the state(s) the system can be in when this use case ends*)

**Map to:** (*List the Identifiers of the user stories that this use case is addressing*)

### <Use Case Description>

**Name:**

**Identifier** (*A unique identifier for this use case, e.g. UC10*)

**Preconditions** (*List the state(s) the system can be in before this use case starts*)

**Basic Course** (*Describe the “normal” processing path, aka, the Happy Path*)

1. Use case begins when …
2. Use case ends when …

**Alternate Course A: Description of the alternate course**

**Condition:** Indicate what happened

1. List the steps

**Post conditions** (*List the state(s) the system can be in when this use case ends*)

**Map to:** (*List the Identifiers of the user stories that this use case is addressing*)

### <Use Case Description>

**Name:**

**Identifier** (*A unique identifier for this use case, e.g. UC10*)

**Preconditions** (*List the state(s) the system can be in before this use case starts*)

**Basic Course** (*Describe the “normal” processing path, aka, the Happy Path*)

1. Use case begins when …
2. Use case ends when …

**Alternate Course A: Description of the alternate course**

**Condition:** Indicate what happened

1. List the steps

**Post conditions** (*List the state(s) the system can be in when this use case ends*)

**Map to:** (*List the Identifiers of the user stories that this use case is addressing*)

### Upload photos and videos use case

**Name: upload photos and videos**

**Identifier** *UC19*

**Preconditions**

1. Event was held and Photos were taken during that event

2.An admin opens his admin profile page.

**Basic Course**

1.Use case begins when an admin navigates to media page

2. Admin starts to upload photos and videos.

3.Use case ends when admin finishes uploading photos and leaves website.

**Alternate Course A: The admin is not authorized to deal with tag requests.**

**Condition:** The admin has no full access and is not a media uploader admin.

A.1 The admin is not allowed to add photos and videos.

A.2 A message appears telling the admin that he has no authorization to add photos and videos.

**Post conditions** (*List the state(s) the system can be in when this use case ends*)

1.Photos and videos are stored in database.

2.photos and videos are uploaded to website

3. users can view new photos and videos.

**Map to:** (*List the Identifiers of the user stories that this use case is addressing*)

### Add finger print use case

**Name: add finger print**

**Identifier**. *UC20*

**Preconditions**

1-Event was held.

2-Exisiting users didn’t upload their own fingerprints.

3.An admin opens his admin profile page.

**Basic Course**

1. Use case begins when admin view list of existing users who don’t have finger print
2. Admin begin to upload their own finger prints after having it from event
3. Use case ends when admin leave website

**Alternate Course A: The admin is not authorized to deal with tag requests.**

**Condition:** The admin has no full access and is not a media uploader admin.

A.1 The admin is not allowed to add finger prints.

A.2 A message appears telling the admin that he has no authorization to add finger prints.

**Post conditions**

1.Figer prints is now added to existing user’s profiles who attended events.

**Map to:** (*List the Identifiers of the user stories that this use case is addressing*)

### View Interviewees use case

**Name: Accept or refuse interviewees**

**Identifier** UC21

**Preconditions**

1-Uesr applied for an Event.

2.An admin opens his admin profile page.

**Basic Course**

1.Use case begins when admin lists the applied users for an event

2. Admin begin to accept or refuse interviewees with giving reason

3.Use case ends when admin hides user lists

**Alternate Course A: Description of the alternate course**

**Condition:** The admin has no full access and is not an Event admin.

A.1 The admin is not allowed to add or refuse interviewees.

A.2 A message appears telling the admin that he has no authorization to add or refuse interviewees.

**Post conditions**

1. Users who applied for event will be informed in their account.

2. Users will have feedback.

3. system will be alerted about those whom have been accepted

**Map to:** (*List the Identifiers of the user stories that this use case is addressing*)

### Create Account for admins use case

**Name: create accounts for admins**

**Identifier** UC22

**Preconditions**

1. Event is held

2.When there is shortage in accounts and new admins are added

3.An admin opens his admin profile page.

**Basic Course**

1.Use case begins when admin navigate to Add new account button

2.Admin Start choosing role for new admin and provide information.

1. Use case ends when Admin navigates to another page.

**Alternate Course A:**

**Condition:** The admin has no full access.

A.1 The admin is not allowed to add new accounts.

A.2 A message appears telling the admin that he has no authorization to add new accounts.

**Alternate Course B:**

**Condition:** Account already exists.

A.1 A message appears telling the admin that account he is trying to create already exists.

**Post conditions** (*List the state(s) the system can be in when this use case ends*)

1.Systen now stores account information about newly added admins

2. Newly added admins can login.

**Map to:** (*List the Identifiers of the user stories that this use case is addressing*)

### Delete account use case

**Name: Delete account use case**

**Identifier** UC23

**Preconditions**

1.Event has finished

2. account has reached its expiry date.

3.Non-working admin.

4.An admin opens his admin profile page.

**Basic Course**

1.Use case begins when Admin list account of other admins

2.Admin starts deleting account of those admin who are not working or if their account reached their expiry date

3.Use case ends when Admin hides list

**Alternate Course A: Description of the alternate course**

**Condition:** The admin has no full access.

A.1 A message appears telling the admin that he is not authorized.

**Post conditions**

1.Some admin account are deleted from systems

2.Those Admins with deleted accounts won’t be able to log in

**Map to:** (*List the Identifiers of the user stories that this use case is addressing*)

### Tag users use case

**Name: Tag users**

**Identifier** UC24

**Preconditions**

1.An admin opens his admin profile page.

2.Some user are not tagged in photo event

**Basic Course**

1. Use case begins when admin navigate to medias
2. Admin start to tag users who attended event in the photos based on their picture stored in their account
3. Use case ends when admin leaves page

**Alternate Course A: Description of the alternate course**

**Condition:** The admin has no full access and is not a Tagger admin.

A.1 A message appears telling the admin that he has no authorization to add tags.

**Post conditions** (*List the state(s) the system can be in when this use case ends*)

1.Systen now store tagged user

2.Tagged user can see media they are tagged in their media.

**Map to:** (*List the Identifiers of the user stories that this use case is addressing*)

## Non-functional Requirements

< These are the constraints imposed by the user or understandable from the user stories and their business. Each requirement should have a unique identifier that can be used as a reference in other documents.>

### <Requirement Description and the user stories ID that this requirement is addressing>

### <Requirement Description>

# High level plan

<Give a high level plan of the schedule of all iterations you expect the project to undertake in a tabular form as follows.>

**Iteration 1:**

|  |  |
| --- | --- |
| User Stories | Estimated Time |
| 1. |  |
| 2. |  |
| **Total Time** |  |

**Iteration 2:**

|  |  |
| --- | --- |
| User Stories | Estimated Time |
| 1. |  |
| 2. |  |
| **Total Time** |  |

**Iteration 3:**

|  |  |
| --- | --- |
| User Stories | Estimated Time |
| 1. |  |
| 2. |  |
| **Total Time** |  |

**Table 1: High Level Plan**

<Please be reasonable in your time estimation. It does not have to be exact, we are here to learn, and now is a good time to learn some time management. *For the software engineering course make each iteration runs for three weeks only. In this course, you will be asked to deliver three iterations only.*>

# Supporting Information

[The supporting information makes the SRS easier to use. It includes: a) Table of contents, b) Index, c) Appendices. These may include use-case storyboards or user-interface prototypes. When appendices are included, the SRS should explicitly state whether or not the appendices are to be considered part of the requirements.]

**Appendix A**

**Traceability Matrix**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **RID** | **US1.1** | **US1.2** |  |  |  |  |  |  | **….** |
| **UC1.1** |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| **NF1.1** |  |  |  |  |  |  |  |  |  |
| **NF1.2** |  |  |  |  |  |  |  |  |  |
| **…** |  |  |  |  |  |  |  |  |  |

<This traceability matrix should provide the relationship between the user requirements (user stories) and the system functional (use cases) and non-functional requirements, such that each check mark indicates a relationship between the user requirements with IDs in column and system requirement with IDs in row. So that any change in a user requirement, from this matrix we can find what system requirements that may be affected by this change.>

**Appendix B**

**Dependability Matrix**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **RID** | **US1.1** | **US1.2** |  |  |  |  |  |  | **….** |
| **US1.1** |  |  |  |  |  |  |  |  |  |
| **US1.2** |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| **…** |  |  |  |  |  |  |  |  |  |

<This dependability matrix should provide the dependency between requirements, such that each check mark indicates a dependency between the requirements with IDs in column and row. So that any change in a requirement or the design work product done later for a requirement is done, from this matrix we can find other requirements that may be affected by this change.>