**Template**

**Outline**  description

**1. Introduction**  the first section responsible to provide a brief overview of the system and its function also its references, objectives and success criteria of the project, its scope and some definitions.

***1.1 Purpose of the system***

***1.2 Scope of the system***

***1.3 objectives and success criteria of the project***

***1.4 Definitions, acronyms and abbreviations.***

***1.5 References.***

**2. Current System** the current system describes if the system will

Replace another system or not and functionality of the system.

**3. Proposed system** the third section documented an overview of the system, its functional requirements (high functionality level of the system) and non-Functional requirements (describe user level requirements).

***3.1 Overview***

***3.2 Functional requirements***

*3.2.1 Sign Up.*

*3.2.2 Log in.*

*3.2.3 Friends*

3.2.3.1 Adding friends

3.2.3.2 Confirming &ignoring friends

*3.2.4 Making posts*

*3.2.5 Liking posts*

*3.2.6 Sharing posts*

*3.2.7 Create page*

*3.2.8 Liking & unlike page.*

3.2.8.1 like page

3.2.8.2. Unlike page

*3.2.9 Create groups*

3.2.9.1 Create public group

3.2.9.2 Create private group

*3.2.10 Join a group*

3.2.10.1 Join public group

3.2.10.2 Join private group

*3.2.11 Leave a group*

*3.2.12 Hash-tag*

*3.2.13 Messages*

3.2.13.1 Send messages.

3.2.13.2 Receive messages.

***3.3 Non Functional requirements***

*3.3.1 Usability*

*3.3.2 Reliability*

*3.3.3 Performance*

*3.3.4 Supportability*

*3.3.5 Implementation*

*3.3.6 Interface*

*3.3.7 Packaging*

*3.3.8 Legal*

**4. Use cases model.**

**1. Introduction**

**1.1 purpose of the system**

The purpose of the project My University is to develop a social network to communicate all the management crew, teaching crew and students in one university together. Only these can register for the network, the purpose is to communicate all the crews and students together in a simple way where teachers can gives there assignments or projects to the students and the students can ask any of teaching or management stuff when they has any questions for or complains about them. Each college in the university can make its page for its students. Also each department in each college can make its own group. Also students can make private groups when they are working at one project to share their thoughts. Also students can add each other as a friend and also doctors or teaching assistants and that make them more sociable with their colleagues and their teaching stuff. Each student, teacher and manager can share his thoughts on his wall he can also like and share others thoughts. Any can make his own group and page.

**1.2 scope of the system**

This software system My university community to facilitate communication among (students, professors, staff) it can be used by maximum 100,000 users.

Specifically, each user (management crew, teaching crew, students ) who signed up to this system con login, share posts, add friend, make chatting, join or create to groups, create or like pages and use hashtag(#).

User can login by his/her ID and password only.

The system also contains a relational database that contains connections, users, and identities .

**1.3 Objectives and success criteria of the project**

The objectives of My University project are to:

1. Providing an application to help in learning process and to communicate easily between management crew, teaching crew and students.

2. To make the students more sociable in the university life when they are just coming out from schools to university.

3. My University application doesn’t need to learn something to be able to use it you can use it from first time without any previous learning.

The project my university is accepted if at least the following success criteria applied:

1. The functional requirements in section 3.2 applied.

1. The program is implemented by java to work in: Windows, mac and Linux operating systems.

**1.4 Definitions, acronyms and abbreviations**

**My University:** the name of the application to be developed.

**Management Crew:** Teaching crew, Students: the type of users in the application.

**1.5 References**

**-**[**https://docs.google.com/viewer?url=http%3A%2F%2Fs3.amazonaws.com%2Fdownload.acadox.com%2Fresource%2F5543%2F9831.pdf%3FAWSAccessKeyId%3DAKIAJEOQKYPKC3CCU5RA%26Expires%3D1415450036%26Signature%3DLaiGJbxccCeS5FMLBV3oNgyNb40%253D**](https://docs.google.com/viewer?url=http%3A%2F%2Fs3.amazonaws.com%2Fdownload.acadox.com%2Fresource%2F5543%2F9831.pdf%3FAWSAccessKeyId%3DAKIAJEOQKYPKC3CCU5RA%26Expires%3D1415450036%26Signature%3DLaiGJbxccCeS5FMLBV3oNgyNb40%253D)

**-**[**https://docs.google.com/viewer?url=http%3A%2F%2Fs3.amazonaws.com%2Fdownload.acadox.com%2Fresource%2F5543%2F9868.pdf%3FAWSAccessKeyId%3DAKIAJEOQKYPKC3CCU5RA%26Expires%3D1415450013%26Signature%3D%252BtyHqYPkD6kKKsLN1XKraDbKlmU%253D**](https://docs.google.com/viewer?url=http%3A%2F%2Fs3.amazonaws.com%2Fdownload.acadox.com%2Fresource%2F5543%2F9868.pdf%3FAWSAccessKeyId%3DAKIAJEOQKYPKC3CCU5RA%26Expires%3D1415450013%26Signature%3D%252BtyHqYPkD6kKKsLN1XKraDbKlmU%253D)

**2. Current System**

There is no current system to replace or develop. It will start for the first time by starting this project.

**3. Proposed System**

**3.1 Overview**

My university is a project in which all the teaching, management crew can communicate with the students throw a social network each of them has his own account then he can have friends make posts. Users can like and share posts by others. They can send messages to each other and use hash-tags. Each user can create his page or group he can also like or unlike pages and join or leave groups.

**3.2 Functional Requirements**

The social network my university support three type of users: users from management crew or teaching crew or students. The users tasks are posting, liking, sharing, creating pages and groups. The users tasks and the behavior of the system leads to the following functional requirements:

**3.2.1 Sign up**

The three types of users can **sign up** using their information: name, email, gender, password and ID from choosing sign up from the welcome page of my university and then by answering the questions asked by the developer when first enter to the site. The web server should get sure that this ID is in the data base with this email which the user has both from his university. If the web server found that the email and ID is right so the user will be able to sign up. Students can have accounts by their name. The management crew and teaching crew have accounts with their name and (√) symbol in front of their name.

**3.2.2 Log in**

When any type of the users log in he can press in log in button in the welcome page of my university if he was signed up before .User must log in using his ID and password and the web server has to check his information in the data base if his information was right then he can log in to My University.

**3.2.3 Friends**

**3.2.3.1 Adding friends**

Every user from any types of users has a list of friends he can add any of the other users to this list by clicking on add friend button of this user account.

**3.2.3.2 Confirming or ignoring friends**

Any friend you add must make a confirmation to be in your list he can ignore being in your list too by clicking show my requests then clicking on button confirm or ignore which will appear next to the users name who adds you. You also can confirm or ignore a friend request.

**3.2.4 Making posts**

Every user (from any types of users) can make posts. Making posts is to write a text to your building wall each of the friends you have in your list could see this post.

**3.2.5 Liking posts**

Every user (from any types of users) can like a post in his wall made by a friend and his posts can be liked by his friends. Web server is responsible to count number of likes for each post and to appear which users like this post by their names.

**3.2.6 Sharing Posts**

Sharing posts is to take a post and share it from a friend as in your wall as if it`s you have the same thoughts. Every user (from any types of users) can share a post in his wall made by a friend and his posts can be shared by his friends. Web server is responsible to count number of Shares for each post and to appear which users Shares this post by their names.

**3.2.7 Create page**

Every user (from any types of users) can create his own page by clicking on a specific button in his wall which has a specific subject where he can make posts their and share his thoughts. Only the creator of the page can post in it. All Users are allowed to see posts in any page, liking and sharing its posts.

**3.2.8 Liking & unlike a page**

**3.2.8.1 Liking a page**

Any user can like any page by clicking on a specific button on this page (like). Web Server is responsible to count number of likes to any page. Users like the page if they want to appear their support to the subject of this page.

**3.2.8.2. Unlike a page**

Any user can unlike a page after liking it if he wants by clicking on the button of unlike in this page and webserver is responsible to remove him from the list of the likers and to decrease the number of likes that is shown by one for each unlike.

**3.2.9 Create groups**

**3.2.9.1 Create public groups**

Every user (from any types of users) can create his own public group by clicking on a specific button in his wall where the users share their thoughts in. Groups can be for making projects or liking something common to share their thoughts about. Public groups all the users can access and like the posts but only members in these groups can post in it.

**3.2.9.2 Create private groups**

Every user (from any types of users) can create his own private group by clicking on a specific button in his wall where some of the users share their thoughts in. Groups can be for making projects or liking something common to share their thoughts about. Private groups only members can access and like the posts and no any other users can access the group.

**3.2.10 Join a group**

**3.2.10.1 Join a public group**

Any user can be a member in the group by sending wish for that by clicking a specific button at the group and the admin (creator) of the group must approve it by clicking on specific button too. Web server is responsible to add, count and list the number of members in each group and it can be shown to all users. Users in public group can make posts and access, like and share the posts in this group.

**3.2.10.2 Join a private group**

Only admin (creator) of the group can add members to his private group by clicking in a button on his group which will make the web server show the users friends list and then he can choose the friend he wants to add him to the group by clicking on him from the list. The web server must add this friend to this group to see and like posts of these groups and to make his own post to this group if he wants.

**3.2.11 Leave a group**

User can leave a public or private group by clicking on a button of leave the group and web server should remove him from the group and decrease the number of members In this group by one (if the group is private he`ll not be able to see posts again).

**3.2.12 Hash-tag**

-Web server should allow users to use hash tag in their posts of their walls and pages. Web server should:

1 - Classify them into Categories. Every hash tag is a category.

2 - Sorting them according to post importance (if it`s written by users of management crew or teaching crew).

Note: Every hash tag should begin with symbol #.

**3.2.13 Messages**

**3.2.13.1 Send messages**

Web server should allow users to send messages by pressing in a button send message and then all the users friends will be shown to him and then he must mark the user(s) he want to start sending message to. Messages can be sent to:

1 - Any other user.

2 - Between 2 users.

3 – Groups contain more than 2 users (3 or more).

**3.2.13.2 Receive messages**

When any other user sent to you a message a hint will occur for you in your wall screen that there is a message when you press on button show messages you could see this message and who has sent it to you.

**3.3 Non Functional Requirements**

**3.3.1 Usability**

**Simple to use:** My University application will be simple to use it will not need any instructions of how to use or how you can access different things in it. You will learn it automatically during access.

**Sharing links:** My University application will allow each user to share links as You tube links or files links (from Google drive or one drive) in his posts as if it isn`t in the application so the profit of the user from the application will be little.

**3.3.2 Reliability**

**Checking**: My university application must be reliable on checking the ID and the email while signing up to not make the students able to access teaching or management stuff accounts.

**3.3.3 Performance**

**Surfing in short time:** My university application should make the user surf the application fast.

**Short checking time:** loading time for My university to check the users ID and password to log in shouldn`t be more than 30 seconds.

**3.3.4 Supportability**

**Friend list:** as you have more friends My University should update no. of your friends during accessing the application and the same for your page or groups My University must react with the likes and members of groups as they increase and decrease.

**Different Pages & groups supportability:** My University can support any type if page or groups any type of user can create it`ll not made the user do a specific type of groups or pages.

**3.3.5 Implementation**

**Programming language:** My University will be implemented using java

**3.3.6 Interface**

**Simple user interface:** My University`s user interface should be easy to use and understandable to all the type of users. It should be understandable for them from the first use without needing to put hints for them to know how to use the application. First when they open the application there will be two buttons to choose from they are sign up and log in if you log in there will be button to show you your pages and other for your groups and two buttons to create page or group and a text place to make a post if you want also your friends posts will appear to you and if you enter any page or group there will be a button to like or unlike this page and to join or leave this group.

**3.3.7 Packaging**

My university application will be available on the web to download it from your university site from one time. The folder the user downloaded will be compressed in it all the files needed for the application to work.

**3.3.8 Legal**

**Cheap price Application:** the application will be sold to universities at cheap price and it will be free for the users to use it in universities. The application will be financed from selling it ti the different universities.

***4. Use case model.***

***4.1 sign up:***

***Use case name: signup***

***Actors: management crew, teaching crew, students.***

***Pre-conditions: actor has an ID in the university.***

***Post-condition: system creates the actor account and login.***

***Flows of events:***

|  |  |
| --- | --- |
| ***User actions*** | ***System actions*** |
| ***1-user enter name, gender, ID, email, password,*** |  |
|  | ***2- system create account and login to it*** |
|  | ***3- ask user to add its profile picture and information about his/her college*** |
| ***4- user enter his/her information and use his/her account*** |  |

***Exceptions:***

***Includes: system checks if this user signed up before or not.***

***Notes and issues: if the user is teaching crew or management crew the system put* (√) symbol in front of his/her name.**

***4.2 login:***

***Use case name: login***

***Actors: management crew, teaching crew, students.***

***Pre-conditions: system checks if the user signed up before or not.***

***Post-conditions: system enters to the user account.***

***Flows of events:***

|  |  |
| --- | --- |
| 1. ***User enters his/her ID and password*** |  |
|  | ***2-System check if the ID and Password is true*** |
|  | ***3-if not the system send message it’s false*** |

***Exception: if user didn’t sign up before ask him/ her to sign up.***

***Includes:***

***Notes and issues:***

***4.3 add friends***

***Use case name :add friends***

***Actors: management crew, teaching crew, students.***

***Pre-conditions:***

***Post-condition:***

***Flows of events:***

|  |  |
| --- | --- |
| ***1-user click on add friend button in the other user account*** |  |
|  | ***2-the system change the button to delete the request*** |
|  | ***3-system add the user account to the friends requests list in the other user account*** |
|  |  |

***Exceptions:***

***Includes:***

***4.4 confirm or ignoring requests:***

***Use case name: confirm or ignoring request***

***Actors: management crew, teaching crew, students.***

***Pre-conditions: the button friend request has one notification if he want to confirm or not***

***Post-condition:if he accept the request his list will increase else his list will bs the same number***

***Flows of events: click on the button of friend request to show who send to me***

|  |  |
| --- | --- |
| ***1-user click on add friend button in the other user account*** |  |
|  | ***2-the system change the button to delete the request*** |
|  | ***3-system add the user account to the friends requests list in the other user account*** |
|  |  |

***Exceptions:***

***Includes:***

***Use case name: makingpost***

***Actors: management crew, teaching crew, students.***

***Pre-conditions: the user wll do to the status text and write what he want***

***Post-condition:the post wll publish on his wall***

***Flows of events: he will open his wall and post what he want***

|  |  |
| --- | --- |
| ***1-user click on add friend button in the other user account*** |  |
|  | ***2-the system change the button to delete the request*** |
|  | ***3-system add the user account to the friends requests list in the other user account*** |
|  |  |

***Exceptions:***

***Includes:***

***Use case name: liking post***

***Actors: management crew, teaching crew, students.***

***Pre-conditions: he click on the post he likes even the person that he like is friend with hm or not,,,if not the post must be public***

***Post-condition: the mark will change to unlike and the numbers of like will increase***

***Flows of events: just if any post like him he will click like on it***

|  |  |
| --- | --- |
| ***1-user click on add friend button in the other user account*** |  |
|  | ***2-the system change the button to delete the request*** |
|  | ***3-system add the user account to the friends requests list in the other user account*** |
|  |  |

***Exceptions:***

***Includes:***

***Use case name: sharing post***

***Actors: management crew, teaching crew, students.***

***Pre-conditions: the user must sure that the post who share it is public to allow to his friends show this post if there arenot friendships between them***

***Post-condition: the post that he shared will post on his wall***

***Flows of events: he will click on the status and write what he want then he post***

|  |  |
| --- | --- |
| ***1-user click on add friend button in the other user account*** |  |
|  | ***2-the system change the button to delete the request*** |
|  | ***3-system add the user account to the friends requests list in the other user account*** |
|  |  |

***Exceptions:***

***Includes:***

|  |  |  |
| --- | --- | --- |
| Use Case ID: | **4.7 Create page** | |
| Use Case Name: | create page | |
| Actors: | crew , teaching crew , students. | |
| Pre-conditions: | actor want page to make posts and share his thoughts. | |
| Post-conditions: | system create page ,and actor start to post on it. | |
| Flow of events: | **User Action** | **System Action** |
| 1- actor click on a specific button in his wall |  |
| 2- name it with specific subject |  |
|  |  |
|  | 3- page created |
| 4- actor starts to write his thoughts |  |
| Exceptions: | Only Actor who create the page can post on it . | |
| Includes: | Users can like ,share and comment on posts . | |
| Notes and Issues: |  | |
| Use Case ID: | 4.8 Liking & unlike a page | |
| Use Case Name: | Liking & unlike a page | |
| Actors: | crew , teaching crew , students. | |
| Pre-conditions: | Number of page's fans is fixed | |
| Post-conditions: | Number of page's fans is decrease or increase | |
| Flow of events: | **User Action** | **System Action** |
| 1- user click on "like" or "unlike" button. |  |
|  | 2- system count the number fans if it decreased or increased. |
| Exceptions: | No exceptions | |
| Includes: | None | |
| Notes and Issues: | None | |
| Use Case ID: | 4.9 Create groups | |
| Use Case Name: | Create groups | |
| Actors: | crew , teaching crew , students. | |
| Pre-conditions: | Need a group to work in lage numbers together | |
| Post-conditions: | Meetings online to do projects and assignments | |
| Flow of events: | **User Action** | **System Action** |
| 1- actor click on create group button. |  |
| 2- name it with specific subject |  |
|  | 3- group created |
| 4- groups' members start work |  |
| Exceptions: | Private groups only members can access and like the posts.  Public groups all the users can access and like the posts but only members in these groups can post in it | |
| Includes: | Private groups.  Public groups. | |
| Notes and Issues: | None | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | **4.10 Join a group** | |
| Use Case Name: | **Join a group** | |
| Actors: | crew , teaching crew , students. | |
| Pre-conditions: | User send add request. | |
| Post-conditions: | Admin or actor may accept or reject it.  Number of groups' members increase | |
| Flow of events: | **User Action** | **System Action** |
| 1- user click on join group button. |  |
|  | 2-admin approve his request. |
|  | 3- system count new member of the group. |
| 4- member start acting with other members. |  |
| Exceptions: | Join a public group Any user can be a member in the group by sending request  Join a private group Only admin of the group can add members to his private group | |
| Includes: | private group  public group | |
| Notes and Issues: | None | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | **4.11 Leave a group** | |
| Use Case Name: | **Leave a group** | |
| Actors: | crew , teaching crew , students. | |
| Pre-conditions: | Number of group's member is fixed. | |
| Post-conditions: | Number of group's member is decrease. | |
| Flow of events: | **User Action** | **System Action** |
| 1- user click on leave group button. |  |
|  | 2-system remove him from group. |
| Exceptions: | if the group is private he`ll not be able to see posts again | |
| Includes: | none | |
| Notes and Issues: | None | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | **4.12 Hash-tag** | |
| Use Case Name: | **Hash-tag** | |
| Actors: | crew , teaching crew , students. | |
| Pre-conditions: | none | |
| Post-conditions: | Database for the hash-tag contents. | |
| Flow of events: | **User Action** | **System Action** |
| 1- user create hash-tag for any thing. |  |
|  | 2- system allow users to use hash tag in their posts |
| Exceptions: | None | |
| Includes: | 1 - Classify them into Categories. Every hash tag is a category.  2 - Sorting them according to post importance (if it`s written by users of management crew or teaching crew). | |
| Notes and Issues: | Every hash tag should begin with symbol # | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | **4.13 Messages** | |
| Use Case Name: | **Messages** | |
| Actors: | crew , teaching crew , students. | |
| Pre-conditions: | Choose friend or user to send him message. | |
| Post-conditions: | Message sent successful. | |
| Flow of events: | **User Action** | **System Action** |
| 1- user choose friend to send him message. |  |
|  | 2- system redirect message for receipts. |
| Exceptions: | None | |
| Includes: | Messages can be sent to:  1- Any other user.  2 - Between 2 users.  3 – Groups contain more than 2 users (3 or more). | |
| Notes and Issues: | none | |