**Student Design Network:**

**Requirements Analysis Document**

**1. Introduction              3**

**1.1 Purpose of the system              3**

**1.2 Scope of the system              3**

**1.3 Objectives and success criteria of the project              3**

**1.4 Definitions, acronyms, and abbreviations              3-4**

**1.5 References              4**

**1.6 Overview              4**

**2. Current system              4**

**3. Proposed system              4**

**3.1 Overview              5**

**3.2 Functional requirements               5-19**

**3.3 Nonfunctional requirements              20**

**3.4 System models              20**

**3.4.1 Scenarios              20**

**3.4.2 Use case model               20**

**3.4.3 Object model               20**

**3.4.4 Dynamic model21-23**

**3.4.5 User interface-navigational paths and screen mock-ups              24**

**4. Glossary              24**

**1. Introduction**

**1.1 Purpose of the system – Problem Statement**

There are many social networks on the Internet today, many are targeted towards keeping in touch, adding applications, meeting new people, or dating.  There are no social networks that are specifically targeted towards Higher Education or helping Employers and Students connect easily with each other in a safe environment away from all of the hype and nonsense of traditional social networks.

Student Design Network should be a place where students, faculty and employers come together to create a network of knowledgeable individuals who will have the opportunity to connect with each other.

Create a social networking websites where students can:

1.      Find jobs in their related fields

2.      Post their resume

3.      Post their portfolio

4.      Create a business and educational network

5.      Find help on a forum

6.      Interact with University staff

University staff members can also join the social network.  They can post information about their classes, help students who need it, and post grades for the students.  Allow the teachers to set up classes where students in those classes can access documents and other helpful information that would be supplementary material to the classroom.

Employers can also log into the website and create a profile page for their business.  They can browse through the students, look at their resumes and portfolios as well as contact the students personally.  Employers will also have the ability to post job listings and target those listings to a certain kind of student.

B. SP ’10

1. Event Calendar: Danays Hechavarria

The purpose of the Event Calendar is to give the students the ability to find out about any new events that are Upcoming in their area.

2. Reply to posted Message: Rahsaan Uldaneta

The purpose is to reply to other messages that are posted by users.

3. Create Message: Shannon Velazquez

The purpose is to post a message in the forums

4.Create Master Category: Dustin Larson

Create administrative categories to help filter categories created by users and keep them organize.

5.Add –A-Picture to Profile: Brandon Bowles

The purpose of this system is to provide a globally unique easy-to-use way for the user to upload his/her own unique avatar to their account profile.

**1.2 Scope of the system – Limitations/Boundaries**

**(ex: target users, i.e. only students from the university)**

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

∙                   Provide an infrastructure for operating a social networking website for students, faculty and employers

∙                   Provide a framework for students to post their resumes and portfolios, interact with other students as well as faculty, and look for jobs

∙                   Provide a framework for faculty to organize information for classes they are teaching and to interact with the students in their classes

∙                   Provide a framework for employers to search for students to join their company

**1.4 Definitions, acronyms, and abbreviations – Object Dictionary**

**User (All) – Student, Staff, Employer, Administrator**

**Logout** **Control** (LogoutOfSystem) allows the user to terminate session with the system.

**Login** **Control** (LoginToSystem) allows the user to connect to the system

**Login** **Form** (LoginToSystem)  container to hold login info to be checked by the server.

**Profile** (LoginToSystem, EditProfile)  Page of displayed information retrieved form the user account.

**SearchForm** (Search)   container for holding search criteria

**SearchControl** (Search) compares search criteria to database

**SearchReport** (Search) displays comparable data based on search criteria

**ProfileForm** (EditProfile)  Page of displayed information retrieved from the user account.

**EditProfileControl** (EditProfile)  Changes information stored in the user profile

**CreateResumeControl** (CreateResume)  Creates  resume on profile

**Resume** (CreateResume, UpdateResume, PrintResume) Work data linked to an account

**ResumeForm** (CreateResume) holds information to be stored in the resume

**createProfileButton** (CreateProfile)  User initiation of the profileControl control

**Profile** Control (CreateProfile, DeleteProfile) Controller for the profile functions

**Account** (All) Contain for holding all of a users information

**Account** **Control** (All) Controller for the Account functions

**infoForm** (CreateProfile)  container for holding the profile information

**Database**/**Information** **Storage** (All) The database that stores the entire websites data

**Confirmation** (All) Confirms all changes to users

**deleteProfileButton** (DeleteProfile) User initiation of the profileControl control

**loginButton** (All) User initiation of the login control

**Resume** **Control** (Update Resume)  Controlls all resume functions

**submitButton**/**postButton** (All) submits information to the database

**updateResumeButton** (Update Resume) User initiation of the resume control

**Message** **Control** (PostMessage) controls all message functions

**Forum** (PostMessage)  container for holding data

**Forum** **Control** (PostMessage) controller for the forum functions

**messageForm** (PostMessage) container for holding messages

**postMessageButton** (PostMessage) User initiation of the postMessage control

**View**/**Edit** **Control** (Print Resume) methods for editing and displaying a resume before it is sent to a printer

**Print** **Control** (Print Resume) method that the website uses to communicate with a printer

**View** (View Profile) methods for viewing all of a users profile information

**Inactive** **Control** (View Profile) The function that log you out for being idle

**1.5 References**

**1.6 Overview**

**2. Current system**

              Non-existing

**3. Proposed system**

**3.1 Overview**

**3.2 Functional requirements – Use cases and Use case diagrams**

|  |  |
| --- | --- |
| Use case name | **CreateProfile** |
| Participating actors | Initiated by User |
| Flow of events | 7.                 1. The User activates the “Create Profile/Account” function on their interface.  8.  2. SYSTEM responds by presenting a form with fields for the user to fill in.  9.  10.             3.  The User fills in the information including username, password, confirm password, and email. The User must also correctly fill in a box with letters that correspond to CAPTCHA image.  Once the form is complete, the user submits the form.  11.  4.  SYSTEM receives the form with the information, verifies that the username and email are not already in the User Database.  If the user does not already exist the SYSTEM inputs the information into the Database.  Then supplies a confirmation page that the user has successfully created an account.  If the user does exist the SYSTEM replies that the user already exist and must change the information. |
| Entry condition | ●        The User does not have an account |
| Exit condition | ●        The User confirms the data for the Database, OR  ●        The User cancels the action, OR  ●        The User receives an error from the SYSTEM. |
| Participating Objects | ●        ProfileForm |

|  |  |
| --- | --- |
| Use case name | **DeleteProfile** |
| Participating actors | Initiated by User |
| Flow of events | 12.             1. The User activates the “Delete Profile” function on their interface.  13.  2. SYSTEM responds by requesting confirmation for deleting the User's profile.    3.  The User either confirms or declines the request from the SYSTEM to delete their profile.  If the User declines, this case is aborted.  Otherwise, continue flow of events.    4.  SYSTEM removes User information from the Database. The User is automatically logged out of SYSTEM and shown a confirmation of deletion. |
| Entry condition | ●        User must have performed LoginToSystem  ●        The User selects the “Delete Profile” function on their interface. |
| Exit condition | ●        SYSTEM removes User information from the Database, OR  ●        The User aborts the case. |
| Participating Objects | ●        DeleteProfileForm |

|  |  |
| --- | --- |
| Use case name | **EditProfile** |
| Participating actors | Initiated by User |
| Flow of events | 14.             1. The User activates the “Edit Profile” function on their interface.  15.  2. SYSTEM responds by presenting a form with their current profile data displayed in the fields. \*\*detail the form  16.  17.             3.  The User changes the desired values of the fields in the form.  Once all the desired changes are made, the User submits the form.  18.  4.  SYSTEM receives the form with changes and notifies the User of the changes made.  19.  20.             5.  The User reviews the submitted data and confirms the changes.  21.  6.  SYSTEM updates the data on the Database. |
| Entry condition | ●        User must have performed LoginToSystem  ●        The User selects to edit their profile. |
| Exit condition | ●        The User confirms the data change(s) for the Database, OR  ●        The User cancels the action, OR  ●        The User receives an error from the SYSTEM. |
| Participating Objects | ●        ProfileForm |

|  |  |
| --- | --- |
| Use case name | **Search** |
| Participating actors | Initiated by User |
| Flow of events | 22.             1. The User types in their desired search string into a text box, choose from a drop down list what they want to search for and submits criteria  23.  2. SYSTEM runs search query in Database for the user's string and returns the results |
| Entry condition | ●        User must have performed LoginToSystem  ●        The User supplies input |
| Participating Objects | ●        SearchForm |

|  |  |
| --- | --- |
| Use case name | **LoginToSytem** |
| Participating actors | Initiated by User |
| Flow of events | 24.             1. The User activates the “Login” function on their interface. User enters information.  Login and Password.  Sends information  25.  2. SYSTEM responds by checking the information Submitted by the User against the database.  If information is a correct match, SYSTEM responds by giving User access to profile |
| Entry condition | ●        The User is not logged into SYSTEM |
| Exit condition | ●        User must have performed LoginToSystem |
| Participating Ojbects | ●        LoginForm |

|  |  |
| --- | --- |
| Use case name | **LogoutOfSystem** |
| Participating actors | Initiated by User  Communicates with SYSTEM |
| Flow of events | 26.             1. The User activates the “Logout” function on their interface.  27.  2. SYSTEM responds by ending the Users session. |
| Entry condition | ●        User must have performed LoginToSystem |
| Exit condition | ●        The User selects “Logout” from their interface. |
| Participating Objects | ●        LogOutButton |

|  |  |
| --- | --- |
| Use Case Name | **ViewProfile** |
| Actor | Initiated by User |
| Events | 1.      User Clicks on View Profile    2.                       The system takes user to their profile page. |
| Entry Condition | The user goes to the website and logs into the system. |
| Exit Condition | User logs out of their account. |
| Quality Requirements | If the user inactive for 10 minutes then the System will log the user out. |
| Participating Objects | ●        Profile |

|  |  |
| --- | --- |
| Use Case Name | **Print Resume** |
| Actor | Initiated by User |
| Events | 1.      User Clicks on View/Edit Resume    2.   System takes user to the View/Edit Resume page.  3.      User clicks on Print Resume link    4.      System runs print command |
| Entry Condition | The user views their Resume once they are logged in. |
| Exit Condition | User receives their printed Resume and exits the page. |
| Quality Requirements | If the user does not click to print the Resume then the print command will not run and the user may continue to modify/view their profile. |
| Participating Objects | ●        Resume |

|  |  |
| --- | --- |
| Use Case name | **CreateResume** |
| Participating Actors | Initiated by User  Communicates with System |
| Flow of events | 28.             1. The User activates the “Create Resume” function on their interface.  29.  2. SYSTEM responds by presenting a form with fields for the user to fill in.  30.  31.             3.  The User fills in the information.  The User must also correctly fill in a box with letters that correspond to CAPTCHA image.  Once the form is complete, the user submits the form.  32.  4.  SYSTEM receives the form with the information. Then supplies a confirmation page to the user.    5. User confirms information  6. System creates resume |
| Entry condition | ∙        User logins in to website  ∙        User accesses the resume page. |
| Exit Condition | ∙        User saves their resume information.  ∙        User receives confirmation that their resume has been created. |
| Quality Requirements | ∙        The User ‘s request to create resume is acknowledged when they click the link to create a new resume.  ∙        The User is able to save the created resume.  ∙        The User is able to later view their newly created resume on their profile. |
| Participating Objects | ●        ResumeForm |

|  |  |
| --- | --- |
| Use Case name | **EditResume** |
| Participating Actors | Initiated by User  Communicates with System |
| Flow of events | 1. The USER activates the update resume link.  2. SYSTEM responds by presenting the USERS’s current resume form with fields for the user to be able to edit.    3. The USER makes the necessary changes. The USER submits the form.  4. SYSTEM receives the form with changes and notifies the User of the changes made.    5.  The USER reviews the submitted data and confirms the changes.                6.  SYSTEM updates the data on the Database. |
| Entry condition | ∙        User logins in to website  ∙        User accesses the resume page. |
| Exit Condition | ∙        User saves their changed information or newly uploaded resume.  ∙        User receives confirmation. |
| Quality Requirements | ∙        The User ‘s request to update resume is acknowledged when they click the link to update a new resume.  ∙        The User is able to save the updated resume.  ∙        The User is able to later view their updated resume on their profile. |
| Participating Objects | ●        Resume |

|  |  |
| --- | --- |
| Use Case name | **CreateMessage** |
| Participating Actors | Initiated by User  Communicates with System |
| Flow of events | 1. THE USER clicks the create new message button.  2. SYSTEM responds by presenting a form with fields for the user to fill in.    3. THE USER fills out their new message in the message form. Including the subjects. And submits the form by pressing the submit button.  4. SYSTEM responds by confirming the USER’s message and posting the new message. |
| Entry condition | ∙        User performs use case : LoginToSystem  ∙        User accesses the forum page. |
| Exit Condition | ∙        User recieves a confirmation that their message has been posted to the forum. OR  ∙        User cancels the action by clicking the cancel button |
| Quality Requirements | ∙        The User is able to later view their posted message on the forum |
| Participating Objects | ∙        messageForm |

­

|  |  |
| --- | --- |
| Use case name | **ViewEventCalendar** |
| Participating actors | Initiated by User  Communicates with System |
| Flow of events | 1.                 1. The User clicks on the “Event Calendar” on their profile page.  2.  2. SYSTEM responds by presenting a calendar with the upcoming events.  3. |
| Entry condition | * User must have performed LoginToSystem |
| Exit condition |  |
| Quality Requirements | ∙                   Have the Google Calendar look |
| Participating Objects | * ViewEvents |

|  |  |
| --- | --- |
| Use Case name | **ReplyMessage** |
| Participating Actors | Initiated by User  Communicates with System |
| Flow of events | 1. THE USER click the reply message link.  2. SYSTEM responds by presenting a form with fields for the user to fill in.    3. THE USER fills out their message in the message form. Including the subjects. And submits the form by pressing the reply button.  4. SYSTEM responds by confirming the USER’s message and posting the reply message. |
| Entry condition | ∙        User performs use case : LoginToSystem  ∙        User accesses the forum page. |
| Exit Condition | ∙        The Reply Message is posted to the form. |
| Quality Requirements | ∙        The font has to be blue |
| Participating Objects | ●        messageForm |

|  |  |
| --- | --- |
| Use case | **Category** |
| Participating actors | Intitiated by Admin communicates with system |
| Flow of events | 1.    Admin clicks on general Admin link  2.    system displays admin tools.  3.    Admin Clicks on createMaster Category link  4.    system ask for name of category  5.    Admin gives name of category  6.    System creates name of category. |
| Entry condition | User must have performed LoginToSystem |
| Exit condition | Admin logs out |

|  |  |
| --- | --- |
| Use case name | **PictureProfile** |
| Participating actors | Initiated by User  Communicates with System |
| Flow of events | 4.      The User logs into SYSTEM of His/Her account and is presented with three different option for selecting picture profile  **5.**    The SYSTEM responds to user’s selection by prompting user to identify picture choice. Via browse and upload through **User computer**, **Web camera or URL**  6.      The User communicates with the SYSTEM by making their choice.  7.      The SYSTEM brings up options to crop  8.      The User finalized cropping and exits by clicking the Upload image button.  9.      The SYSTEM saves the user’s selected image  10.  The SYSTEM closes picture upload dialogue window. |
| Entry condition | ●        User must make selection as to which image he/she wants |
| Exit condition | ●        The User is satisfied with conditions made  ●        The User quits image upload function by clicking on the exit. |
| Participating Objects | Picture profile |

**3.3 Nonfunctional requirements**

∙        System can be scalable.  It can grow as needed with the growth in popularity and the rise in the student population.

∙        Extensibility.  The administrator must be able to trouble shoot any problems that occur with student logins or system failures as well respond to inquiries provided by the users within a few hours.

∙        Ease of use.  The social network is using a user interface standard familiar to computer users of all ages and a help menu/FAQ’s page would also be provided.

**3.4 System models**

**3.4.1 Scenarios**

**3.4.2 Use case model**

**3.4.3 Object model**

**3.4.4 Dynamic model**

**3.4.5 User interface-navigational paths and screen mock-ups**

              N/A

**4. Glossary**

**CreateProfile** – Allows Users to create a profile with their information displayed.

**DeleteProfile** – Allows Users to remove their profile from the SYSTEM.

**EditProfile** – Allows Users to edit the information displayed in their profile.

**Search** – Allows Users to scan the SYSTEM for any relevant information based on keywords of their choosing.

**LoginToSystem** – Allows Users with an account to enter the SYSTEM and view any available information, including their personalized profile.

**LogoutOfSystem** – Allows Users to exit their personalized profile of SYSTEM.

**ViewProfile** – Allows Users to view the information on their own or another User’s profile.

**PrintResume** – Allows Users to print a hardcopy of their own or another User’s resumes

**CreateResume** – Allows Users to create a resume to be associated with their profile.

**EditResume** – Allows Users to edit the information displayed on their resume.

**PostMessage** – Allows Users to display a message in the forum section of the SYSTEM.

**ViewEventCalendar** – Allows Users to view new events.

**ReplyMessage** – Allows Users to reply to other messages that are posted by users.

**CreateMessage** – Allows Users to post a message in the forums.

**Category** – Allows Users to create administrative categories to help filter categories created by users for organization.

**PictureProfile** – Allows Users to upload pictures to their account profile.