**Computer Science and Engineering**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PolyFace Dashboard**

**CS 4523 - Software Design Description (SDD)**

**Version 1.1**

Document Number: SDD-001

Team Number: B5

Team Members:

Basia Bowens 0409049 ([bbowen01@students.poly.edu](mailto:bbowen01@students.poly.edu))

Wendy Lau 0402381 ([wlau01@students.poly.edu](mailto:wlau01@students.poly.edu))

Wayne Jones 0406890 ([wjones01@students.poly.edu](mailto:wjones01@students.poly.edu))

**Review And Approvals**

|  |  |  |  |
| --- | --- | --- | --- |
| **Printed Name and Title** | **Function( Author, Viewer, Approval)** | **Date** | **Signature** |
| Professor Strauss | Author | April 9th, 2013 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Revision Level**

|  |  |  |
| --- | --- | --- |
| **Date** | **Revision Number** | **Purpose** |
| March 5th, 2013 | 1.0 | Initial Release |
| April 9th, 2013 | 1.1 | Final Release |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Table Of Contents**

**1. INTRODUCTION ................................................................................................................................ 4**

**1.1 PURPOSE ......................................................................................................................................... 4**

**1.2 SCOPE ............................................................................................................................................. 4**

**1.3 IDENTIFICATION.............................................................................................................................. 4**

**1.4 DOCUMENT SUMMARY ................................................................................................................... 4**

**1.5 SYSTEM OVERVIEW........................................................................................................................ 4**

**1.6 DOCUMENT OVERVIEW................................................................................................................... 4**

**2. REFERENCE DOCUMENTS............................................................................................................. 5**

**3. SYSTEM WIDE DESIGN DECISIONS.............................................................................................5**

**3.1 SOFTWARE COMPONENT ARCHITECTURAL DESIGN........................................................................ 5**

**3.2 SOFTWARE ARCHITECTURE GENERAL DESCRIPTION......................................................................5**

**3.3 SOFTWARE ITEM COMPONENTS...................................................................................................... 6-7**

**3.4 COMPONENT INTERFACE IDENTIFICATION...................................................................................... 7**

**3.5 SOFTWARE COMPONENT CONCEPT OF EXECUTION ........................................................................ 7**

**4. SOFTWARE ITEM DETAILED DESIGN........................................................................................ 8**

**4.1 STRUCTURE..................................................................................................................................... 8**

***4.1.1 Software Unit Detailed Design.................................................................................................. 8***

**4.2 STATIC RELATIONSHIP OF SOFTWARE UNIT ................................................................................... 9**

***4.2.1 Run-time Object Instances......................................................................................................... 9-11***

**4.3 BEHAVIOR....................................................................................................................................... 11**

***4.3.1 Interaction Diagrams................................................................................................................. 11***

***4.3.2 Collaboration Diagrams............................................................................................................ 11***

***4.3.3 Activity Diagrams...................................................................................................................... 12***

**4.4 CONCEPT OF EXECUTION................................................................................................................ 12**

**4.5 INTERFACE DESIGN......................................................................................................................... 12**

***4.5.1 Interface Identification and Diagrams...................................................................................... 12***

***4.5.2 Unique identifier of Interface .................................................................................................... 12***

**6. DEPLOYMENT ARCHITECTURE.................................................................................................. 13**

**6.1 PHYSICAL DEPLOYMENT ARCHITECTURE DIAGRAM...................................................................... 13**

**7. DICTIONARIES................................................................................................................................... 13**

**8. SOFTWARE ITEM COMPUTER RESOURCE UTILIZATION.................................................. 14**

**9. REQUIREMENTS TRACEABILITY................................................................................................ 14**

**9.1 SOFTWARE COMPONENT-LEVEL REQUIREMENTS TRACEABILITY .................................................. 14**

**10. SYSTEM DESIGN TESTING......................................................................................................... 15**

**13. APPENDICES................................................................................................................................... 15**

**13.1 DICTIONARIES...................................................................................................................................... 15**

**13.2 UML DIAGRAMS, IF NOT INCLUDED IN THE BODY OF THE DOCUMENT ................................................. 15**

**13.3 SCHEDULE TRACKING.......................................................................................................................... 15**

**13.4 DEFECT TRACKING............................................................................................................................... 15-16**

**1. INTRODUCTION**

**1.1 Purpose**

The purpose of this Software Design Document is to define the content and work that will go into the PolyFace Dashboard. This document will outline everything to do with the design of the PolyFace system from the component architectural design, the concept execution and numerous diagrams. This document will display the system’s content characteristics as well as the thought process behind that execution.

**1.2 Scope**

The scope of this document extends to describing the system’s characteristics. The document’s scope also extends to describe the system’s behavior and all its architectural components, as well as any other components related to the document and the system such as collaborations of any sort. The document also goes into implementation architecture. There is also a dictionary included in the document.

**1.3 Identification**

The name of the system is: PolyFace or The PolyFace Dashboard. This document pertains to the first release of the system as none have been completed to date. Any documents related to this document are listed in section two under the title of Reference Documents.

**1.4 Document Summary**

This document serves to give the technical and in depth view of the PolyFace Dashboard system. The intended audience for this document includes the team that is putting together the system, the advisor, and any individual that might be making edits to the system in the future. This document does not cater to the client as it goes very in depth on the technical aspect of things.

**1.5 System Overview**

The PolyFace Dashboard is a system that will allow the NYU-Poly community to easily view different types of events and opportunities that are occurring on and off campus for NYU-Poly community members. The system also allows the users of the systems to make posts to the dashboard, and possibly become subscribed members.

**1.6 Document Overview**

This document will outline everything to do with the design of the PolyFace system from the component architectural design, the concept execution and numerous diagrams. This document will display the system’s content characteristics as well as the thought process behind that execution.

**2. REFERENCE DOCUMENTS**

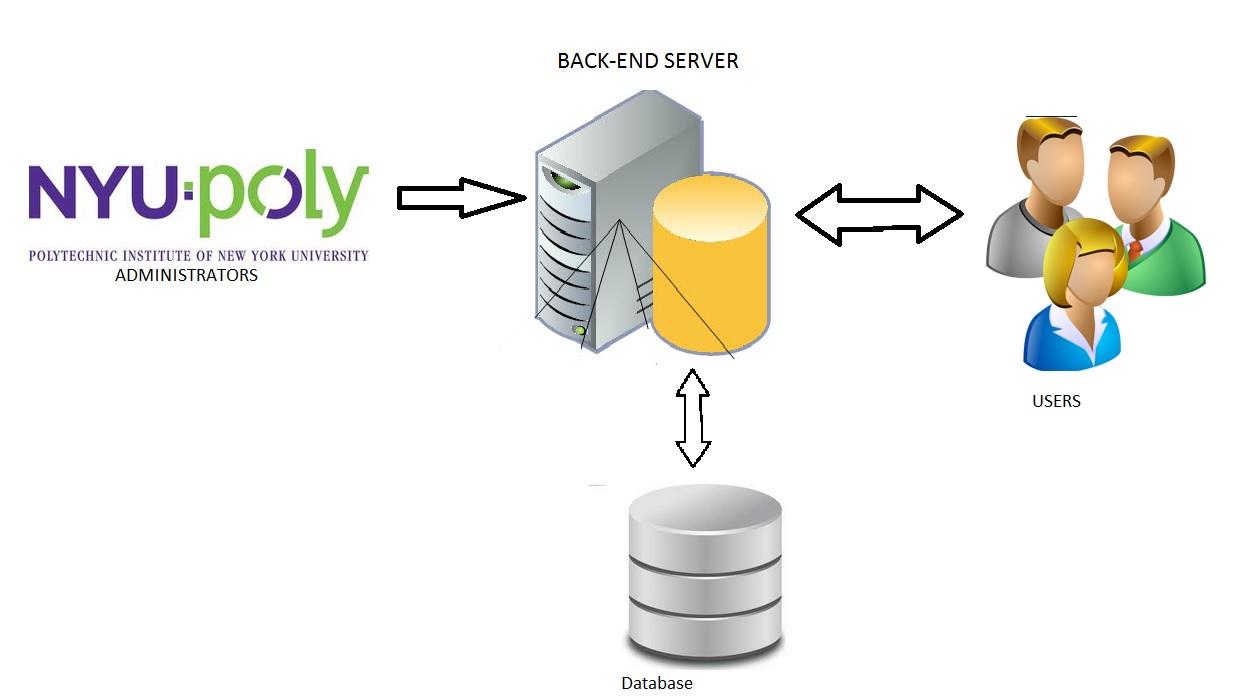
PolyFace Project Proposal, Version 1.1, February 4, 2013

PolyFace, RAS, Version 1.1, February 18, 2013

PolyFace, SPMP, Version 2.1, February 21, 2013

**3. SYSTEM WIDE DESIGN DECISIONS**

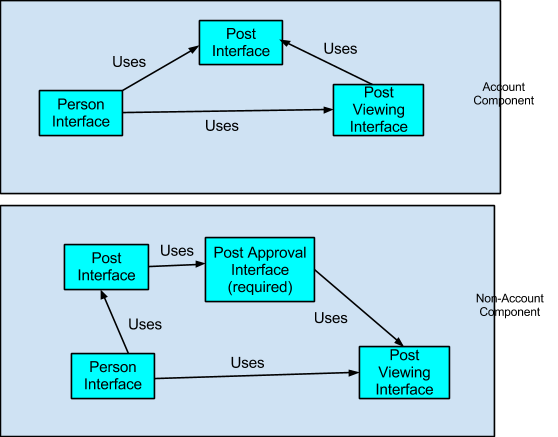
**3.1 Software Component Architectural Design**



**3.2 Software Architecture General Description**

PolyFace is designed for the NYU-Poly community to have simplified and convenient access to events, reminders, social outings, etc. The dashboard will have a user friendly web interface where users can easily access information to their desire. It will be well organized with simplicity so it can be dubbed an useful product to the students of NYU-Poly.

**3.3 Software Item Components**



Users:

1. Account Component : If users hold an account, they are eligible to post without having the overhead of Administrators

* Person Interface (required)
  + uses the majority of the other interfaces
* Post Interface
  + used by the person interface
* Post Viewing Interface

1. Non-Account Component: Users have limited capabilities as compared to users with an account

* Person Interface (required)
* Post Interface
* Post Approval Interface
* Post Viewing Interface

Database: will store information collected from users for their personal profiles; General information of the schools events will be kept track

NYU-Poly Administrators: Consolidates all NYU-Poly community events and information from respective areas to update the dashboard

Back- End Server plays the crucial role. Information will go through the server as it communicates with users and the dashboard’s respective database.

**3.4 Component Interface Identification (See section 4)**

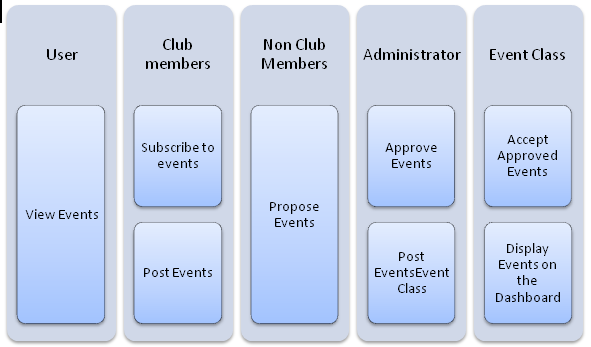
**3.5 Software Component Concept of Execution**

When a user utilizes the dashboard, the database will execute their accounts and provide them with their appropriate capabilities around the website. A user with an account component would execute their capabilities and rights to work around the dashboard. They will be able to read, post and modify. On the other hand, user without accounts will only have permissions to a certain extent. Certain options that may be limitations include posting without the overhead of administrators or ability to view private events subjected to only accounted users.

**4. SOFTWARE ITEM DETAILED DESIGN**

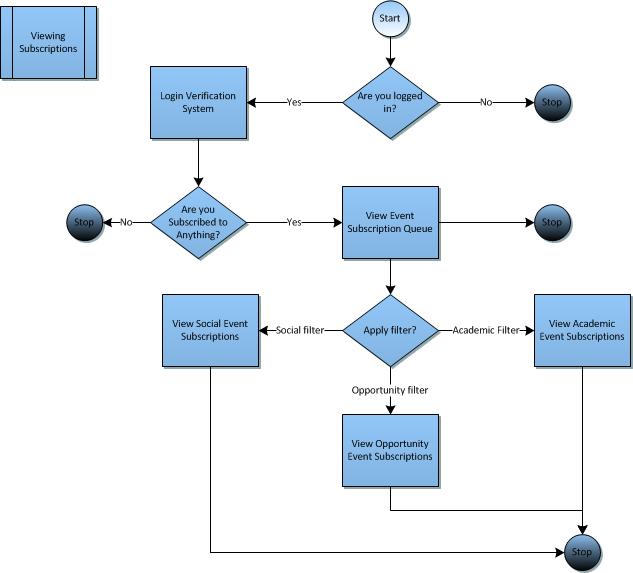
**4.1 Structure**

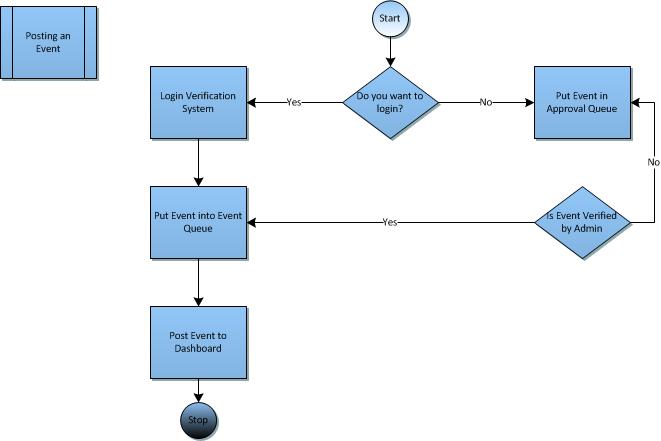
**4.1.1 Software Unit Detailed Design**

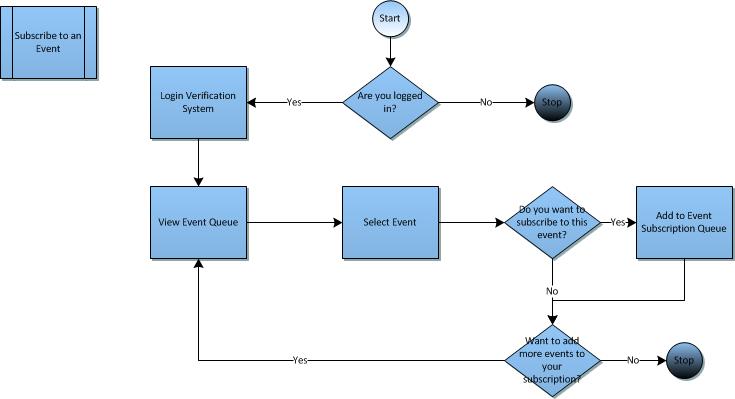
****

**4.2 Static Relationship of Software Unit**

**4.2.1 Run-time Object Instances**

**Key Event Diagrams**



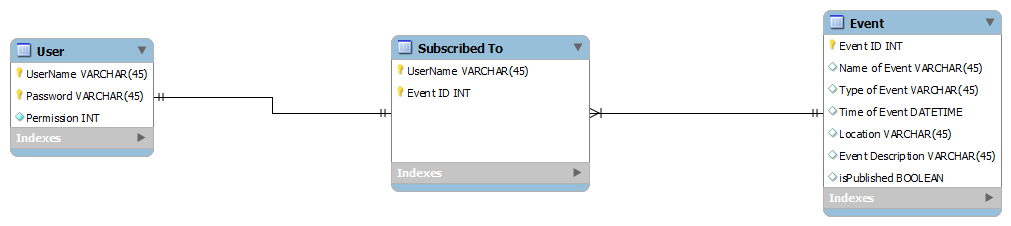


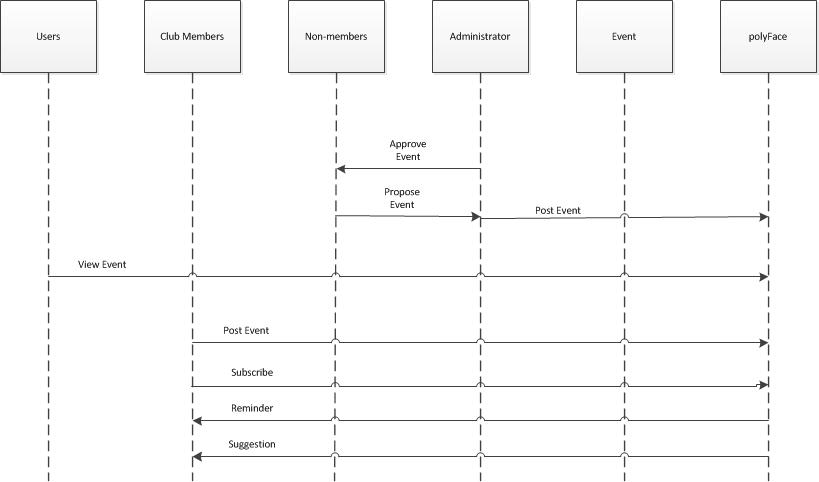
**Key Event Dictionary**

* 1. Class Users: Models all users that the system interacts with
  2. Class Members: Models users that are capable of logging in.
  3. Class Guests: Models users that have no log in information.
  4. Class Posts: Models all submissions and events.
  5. Class Admin: Models the system administrator.
  6. Class polyFace: Models the system itself.
  7. Method submit: This function allows members to directly post event information to the system. The event information is sent as a string to the polyFace system.
  8. Method propose: This function allows non-members(guests) to send their event information to the administrator for approval. Once approved, the information will show on the system.
  9. Method approval: This function provides the administrator with the capability of approving or denying the event proposal of a guest. It returns true if the proposal is approved and false otherwise.
  10. Method login: This function sends the Id and password of the member to the PolyFace system for verification.
  11. Method verify: This function verifies that the Id and password it is passed, exists in the system. If the log in information exists, then the function returns true, otherwise it returns false.
  12. Method view: This function is shared by all users and allows them to view the event information found in class Posts.
  13. Attribute status: This attribute declares whether or not a submission to the system has been approved. It holds a default value of false.
  14. Attribute active: This attribute declares whether or not an event has passed. If the date of the event has passed then this variable will be set to false, indicating that it should not be active in the system.
  15. Attribute type: This attribute declares what category of event the submission is (Academic, Social, Opportunity, etc).
  16. Attribute id: This attribute holds the value of the member’s user name.
  17. Attribute password: This attribute holds the value of the member’s password.

**4.3 Behavior**

**4.3.1 Interaction Diagrams**

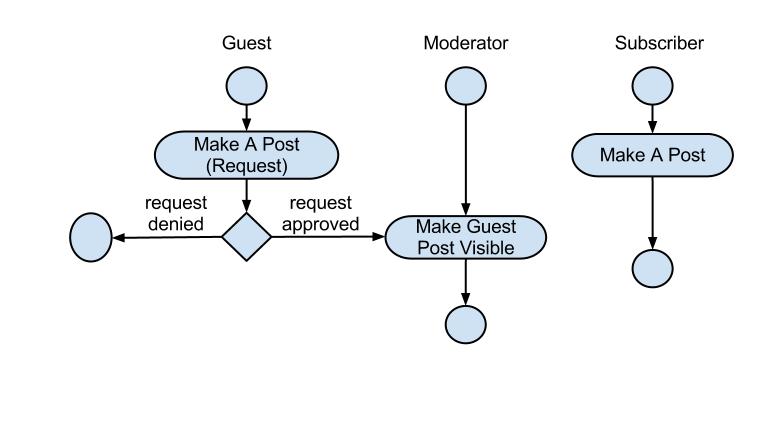
****



**4.3.2 Collaboration Diagrams**

There are no collaboration diagrams as of this point in our software design phase. Collaboration Diagrams will be included in the next iteration of this Software Design Document.

**4.3.3 Activity Diagrams**

****

**4.4 Concept of Execution**

The concept of the execution of this dashboard is to make the dashboard as easily navigable as possible, as well as to make the dashboard as intuitive as possible. The diagrams in this document display the different architectures and relationships that are pivotal in order for this concept to be completed correctly. The concept of execution is fairly simple and, in the team following through on all the software designs, will be completed as such.

**4.5 Interface Design**

**4.5.1 Interface Identification and Diagrams**

Database – Store event information and over time make room in the database by clearing old events. Also we will query the database for information and update the information, if needed.

GUI – To administer a quality user visual experience. Parts of the GUI website will be reusable such as the header and navigation section of the website.

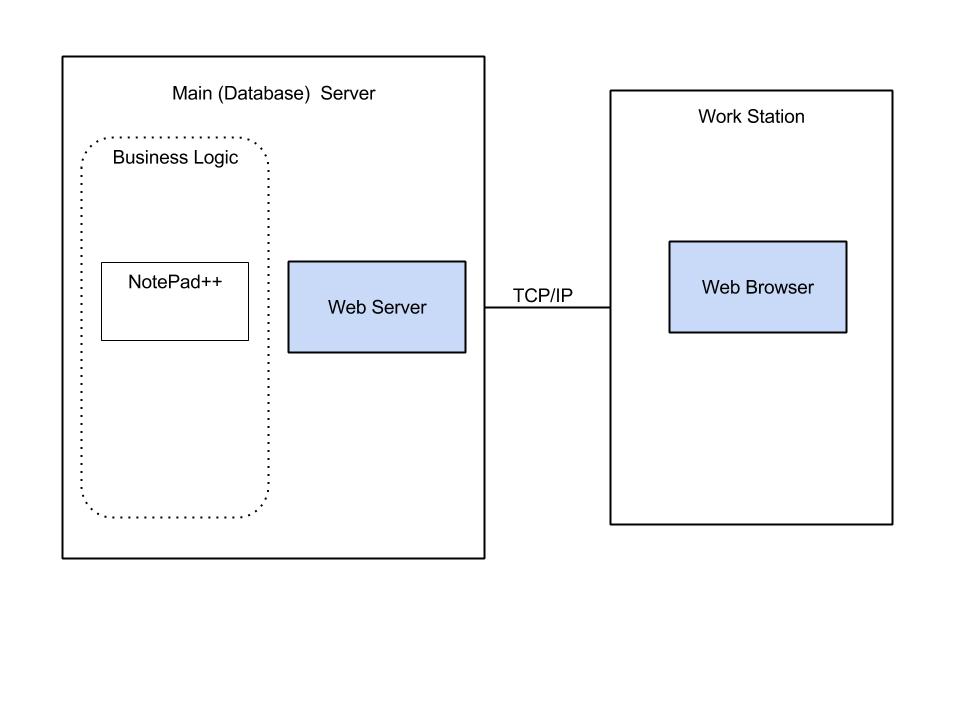
Server – To host files for the PolyFace Dashboard

**4.5.2 Unique identifier of Interface**

There are no additional interfaces for our system.

**6. DEPLOYMENT ARCHITECTURE**

**6.1 Physical Deployment Architecture Diagram**

****

**7. DICTIONARIES**

**User -** the person using the dashboard

**Username -** the name the user uses to login

**password -** the unique each user uses to login

**permission -** the type of allowance for each user

**Event -** an occurrence going on on campus

**EventID -** the unique id code for each event

**TypeOfEvent -** the type of event occurring i.e. Academic, Social , Reminder etc

**PlaceOfEvent-** location of event

**TimeOfEvent -** time of event

**EventDescription -** contains the details of the event

**IsPublished -** tells admin. if the event has been approved or not

**SubscribedTo -** contains all subscribed to events

**Username -**  name of user subscribed to the event

**EventID-** id code of event subscribed to

**8. SOFTWARE ITEM COMPUTER RESOURCE UTILIZATION**

All the software resources that we require are open-source materials. We will customize this software to build the front-end and back-end of the PolyFace Dashboard.

**9. REQUIREMENTS TRACEABILITY**

**9.1 Software Component-Level Requirements Traceability**

|  |  |  |
| --- | --- | --- |
| Requirements | Requirement Number | Use Case |
| Given full posting privileges as a club executive board membership holder | REQ - 001 | 7.2.0-1 |
| Freedom ONLY to post about that particular club’s events as a club executive board membership holder | REQ-002 | 7.2.0-1 |
| Post don’t require Administrative Monitoring as a club executive board membership holder | REQ - 003 | 7.2.0-1 |
| Given limited posting privileges if user does not hold club membership | REQ-004 | 7.2.0-2 |
| Can send a post request but it must be approved by Administrators if user does not hold club membership | REQ-005 | 7.2.0-2 |
| Users without club membership can post about any event, as long as it is relevant information | REQ-006 | 7.2.0-2 |
| The Academic Record of Events will update users on upcoming study sessions, tutoring sessions, project opportunities, review sessions, and stress relief related events. | REQ-007 | 7.2.0-3.1 |
| For further opportunities, the Academic Record of Events also be posting dates of career fairs, info sessions, mock interviews,internship opportunities and networking events | REQ-008 | 7.2.0-3.2 |
| The system administrator will post social events according and relating to holidays will be posted | REQ-009 | 7.2.0-3.2 |
| Student gathering including club events, meets and greets,networking opportunities, movie nights, etc. will be notified on the dashboard by the system administrator | REQ-010 | 7.2.0-3.2 |
| Reminders will consist of all categories, but mostly for academic calendar specific reminders. | REQ-011 | 7.2.0-4 |
| The dashboard will notify the users of all events at which there will be free food | REQ-012 | 7.2.0-6 |

**10. System Design Testing**

Developers will perform unit testing on the source code for an individual unit class or method. Changes and modifications will be made appropriately to ensure faults will not occur. Unit testing will be done to demonstrate that every method in the class works correctly.

System testing will include a separate test for each functional and nonfunctional requirement included in the SRS. We will include scenario testing where we act as a user utilizing the dashboard as members and as non members. Test cases will include inputting valid login sessions, creating and commenting on posts, retrieving information on events, etc. If the system is not performing as desired or errors are occurring, further testing must be done. Each test will be run and logged for future reference.

**13. APPENDICES**

**13.1 Dictionaries**

(Included in documentation)

**13.2 UML diagrams, if not included in the body of the document**

(Included in documentation)

**13.3 Schedule Tracking**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Artifact/ Deliverable** | **Who** | **Estimated** | **Actual** | **Difference** |
| SDD | Wendy Lau | 2 hours | 2 hours | 0 hours |
| SDD | Basia Bowens | 3 hours | 3 hours | 0 hours |
| SDD Compilation | Wayne Jones | 4 hours | 5 hours | 1 hours |

**Cumulative Totals**

|  |  |  |  |
| --- | --- | --- | --- |
| **Who** | **Estimated** | **Actual** | **Difference** |
| Basia Bowens | 5½ hours | 4 ½ Hours | 1 hour |
| Wayne Jones | 3 hours | 5 hours | 2 hours |
| Wendy Lau | 3 hour | 2hours | 1 hours |

**13.4 Defect Tracking**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Artifact/ Deliverable** | **Who** | **Estimated** | **Actual** | **Difference** |
| SDD | Wendy Lau | 2 |  |  |
| SDD | Basia Bowens | 3 |  |  |
| SDD | Wayne Jones | 3 |  |  |

**Cumulative Totals**

|  |  |  |  |
| --- | --- | --- | --- |
| **Who** | **Estimated** | **Actual** | **Difference** |
| Basia Bowens | 3 |  |  |
| Wayne Jones | 3 |  |  |
| Wendy Lau | 2 |  |  |