Virtual Classroom

Software Requirements Specification

COP 4331C, Fall, 2015

## **Modification History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Who | Comment |
| V 0.0 | 10/1/2015 | J. Casserino | Initial Draft |
| V 1.0 | 10/6/2015 | J. Bender | Added Assumptions, |

Team Name: Group 26

Team Members:

Joseph Bender [jbender94@knights.ucf.edu](mailto:jbender94@knights.ucf.edu)

Joshua Casserino [Joshua.casserino@knights.ucf.edu](mailto:Joshua.casserino@knights.ucf.edu)

Chad Armstrong [chad.armstro@knights.ucf.edu](mailto:chad.armstro@knights.ucf.edu)

Miles Friedman [milesfriedmanfl@gmail.com](mailto:milesfriedmanfl@gmail.com)

Contents of this Documents

Introduction.............................................................................................................................

Software to be Produced..............................................................................................

Reference Documents..................................................................................................

Applicable Standards...................................................................................................

Definition, Acronyms, and Abbreviations..................................................................

Product Overview...................................................................................................................

Assumptions................................................................................................................

Stakeholders................................................................................................................

Event Table..................................................................................................................

Use Case Diagram.......................................................................................................

Use Case Descriptions.................................................................................................

Specific Requirements.............................................................................................................

Functional Requirements.............................................................................................

Interface Requirements................................................................................................

Physical Environment Requirements..........................................................................

Users and Human Factors Requirements....................................................................

Documentation Requirements.....................................................................................

Data Requirements......................................................................................................

Resource Requirements...............................................................................................

Security Requirements................................................................................................

Quality Assurance Requirements................................................................................

Supporting Material.................................................................................................................

**Introduction - Josh**

**Software to be Produced**

<one paragraph to identify the software produced(s) to be produced.. Refer the reader to the reference documents for more information>

**Reference Documents**

* Concept of Operations <include link here>
* Project Plan <include link here>
* <any other relevant documents; include full reference information or link>

**Applicable Standards**

<You do not have to repeat the standards included in the project plan. This is where you cite any standards that are specific to the system requirements.>

**Definition, Acronyms, and Abbreviations**

<include any that are needed to read this document (or "None" if document is self-explanatory and no acronyms or abbreviations will be used>

**Product Overview**

**Assumptions**

The mobile iOS application will interface with a PHP web-service that will query the MySQL database setup to house all of the applications data entities and relationship data.

The iOS app will be deployed to an iPhone 6 running iOS 9.0.2 for purposes of demoing the application. The iPhone 6 features a 4.7” LED multi-touch display, an A8 chip with 64-bit architecture, and a capacity of 64 GB. The system user interface does inspire from user interface design concepts and process flows within Instructure’s Canvas iOS application.

**Stakeholders - Josh**

<A stakeholder is anyone who has an interest in the software to be developed. For example, the customer, the various classes of users, applicable regulatory agencies, ... List each category of stakeholder and give a phrase or a sentence to describe their interest or concerns>

**Event Table - Josh**

<An event table identifies all the external events to which the software must respond. This is a first step in determining the required overall system functionality. The event list should be consistent with the context diagram and the interest of each stakeholder. Make sure that exceptions are considered.>

<Use the following table format :>

|  |  |  |  |
| --- | --- | --- | --- |
| Event Name | External Stimuli | External Responses | Internal data and state |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Use Case Diagram - Miles**

<Include a use case diagram here. It should be consistent with all the above work For reference, see materials presented in class.Your text also has a little information.>

**Use Case Descriptions - Miles**

<Briefly describe each use case included in the above diagram. Make sure that exceptions are considered. For reference, see materials presented in class. class.Your text also has a little information >

**Specific Requirements**

**Functional Requirements**

|  |
| --- |
| No: 001 |
| Statement: The system shall allow users to login to the system using their email address and created password. |
| Source: Login |
| Dependency: 002 |
| Conflicts: None |
| Supporting Materials: *insert here* |
| Evaluation Method: A test case is fully executed where an account is created with an email address and password. Then the password and email address are input at the Login screen, and the application presents the user with their profile home page. |
| Revision History: J. Bender, 10/6/2015, Initial Draft |

|  |
| --- |
| No: 002 |
| Statement: The system shall allow a student to create an account by providing their first name, last name, school, email address (as login ID), and a valid password against the security constraints of must contain 8 characters, must contain at least one numeric character, and must contain at least one uppercase letter. This information will be saved and the user will receive validation that the account creation was either successful or not. |
| Source: Student Account Creation |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: |
| Evaluation Method: After the input information is received, either a new account is created (if one does not already exist under the provided email address) and the information is stored in the database, and the user receives a message on screen verifying account creation. If an account is already in place in the database under the email address provided, then the user shall receive an onscreen message that the account could not be created because the email is already in use. |
| Revision History: J. Bender, 10/6/2015, Initial Draft |

|  |
| --- |
| No: 003 |
| Statement: The system shall allow an Instructor to create an account by providing their first name, last name, school, email address (as login ID), and a valid password against the security constraints of must contain 8 characters, must contain at least one numeric character, and must contain at least one uppercase letter. This information will be saved and the user will receive validation that the account creation was either successful or not. |
| Source: Instructor Account Creation |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: |
| Evaluation Method: After the input information is received, either a new account is created (if one does not already exist under the provided email address) and the information is stored in the database, and the user receives a message on screen verifying account creation. If an account is already in place in the database under the email address provided, then the user shall receive an onscreen message that the account could not be created because the email is already in use. |
| Revision History: J. Bender, 10/6/2015, Initial Draft |

|  |
| --- |
| No: 004 |
| Statement: The system shall allow students to browse courses available at their school and add select ones to their profile by course name and ID. |
| Source: Student Course Enrollment |
| Dependency: 001, 002 |
| Conflicts: None |
| Supporting Materials: |
| Evaluation Method: The system displays the available courses in a scrollable window on screen and also allows the users to enter in the course name and ID into a text field to search for the course. A course will be able to be selected and added to the user’s profile. These changes should be reflected in the user’s home page, and within the database. |
| Revision History: J. Bender, 10/6/2015, Initial Draft |

|  |
| --- |
| No: 005 |
| Statement: The system shall allow instructors to create a course they are instructing at their school of employment. The input should include the course name and ID, the ability to upload files, and the ability to create assignments. |
| Source: Instructor Course Creation |
| Dependency: 001, 003 |
| Conflicts: None |
| Supporting Materials: |
| Evaluation Method: The system requires a course name and ID to be input, and all aspects described above are able to be input for the course. All of these updates and changes for the course should reflect on the Instructor’s home page and in the database. |
| Revision History: J. Bender, 10/6/2015, Initial Draft |

|  |
| --- |
| No: 006 |
| Statement: The system shall allow student users to view and download files for each course. |
| Source: Students Viewing/Downloading Course Files |
| Dependency: 001, 003, 004 |
| Conflicts: None |
| Supporting Materials: |
| Evaluation Method: This requirement does not require any input. The system will show a list of files for a specific course, and a user should be able to download a file (docx, pdf, pptx) and view the content on the device. |
| Revision History: J. Bender, 10/6/2015, Initial Draft |

|  |
| --- |
| No: 007 |
| Statement: The system shall allow student users to view and submit assignments for each course. |
| Source: Student Assignment View/Submit |
| Dependency: 001, 003, 004 |
| Conflicts: None |
| Supporting Materials: |
| Evaluation Method: This requirement will show the user a screen displaying all of the current assignments for the course. A file of the specified type will be able to be submitted electronically. |
| Revision History: J. Bender, 10/6/2015, Initial Draft |

|  |
| --- |
| No: 008 |
| Statement: The system shall allow student users to view their current grades for a course. |
| Source: Student View Grades |
| Dependency: 001, 003, 004 |
| Conflicts: None |
| Supporting Materials: |
| Evaluation Method: This requirement does not require any input. The system will show a list tabular view displaying all of the current grades the student has received from the instructor for each assigned item in the course. |
| Revision History: J. Bender, 10/6/2015, Initial Draft |

|  |
| --- |
| No: 009 |
| Statement: The system shall allow student users to create, view, and post on discussion boards |
| Source: Student Discussion Post |
| Dependency: 001, 003, 004 |
| Conflicts: None |
| Supporting Materials: |
| Evaluation Method: This requirement will provide a screen to view all existing discussions in the course, and provide a text field for users to append a comment on the discussion topic. The system will also provide a functionality to create an entirely new discussion thread with a title and other pertinent details. |
| Revision History: J. Bender, 10/6/2015, Initial Draft |

|  |
| --- |
| No: 010 |
| Statement: The system shall allow instructor users to view assignment submissions and provide a grade for them. |
| Source: Instructor Grade Assignments |
| Dependency: 001, 003, 005, 007 |
| Conflicts: None |
| Supporting Materials: |
| Evaluation Method: The system will allow for an instructor account to view all of the submissions for an assignment, and provide a grade for the assignment. |
| Revision History: J. Bender, 10/6/2015, Initial Draft |

**Interface Requirements**

|  |
| --- |
| No: 011 |
| Statement: The system will interface with a PHP web-service that helps query the MySQL database where all data, files, and other entity information relevant for the system will be stored. |
| Source: Login |
| Dependency: Software System, Database, and Php Web Service |
| Conflicts: None |
| Supporting Materials: *insert here* |
| Evaluation Method: All data items are accurately sent, fetched, and received between the database and the software system.   * Data input items include: usernames, passwords, names, files for assignments, grades, discussion post content, course content * Data output items include: grades, files, course names, assignment preview content * Data Type Domains   + Usernames : Unique String containing an email address   + Password : String of at least 8 characters containing at least one number and one uppercase letter   + Names : String of characters limit of 30 characters   + Course Name: String of 20 characters   + Course ID: String containing the three letter prefix code and 4 digit number   + Files : range of file types supported, pdf, doc, docx, ppt, and pptx, txt, mp3, mp4   + Grades : Unsigned double numeric value   + Discussion Posts: Strings limited to 500 characters * Data Accuracy: Most data items will be user input and the accuracy is dependent upon the user input. Passwords and usernames will be strictly required to adhere to their domains. * Data will be sent constantly as users are navigating through the application, because specific content needs to be loaded per specific user. * The data will be loaded synchronously. Any changes will be recognized across the entire database in real time. |
| Revision History: J. Bender, 10/7/2015, Initial Draft |

**Physical Environment Requirements - Josh**

< Describe the environment in which the software must run. Physical environment requirements include:

* type of equipment on which the software must run
* location of the equipment
* environmental considerations: temperature, humidity, ...
* ...>

**Users and Human Factors Requirements - Josh**

<Describe the users and their constraints:

* What different types of users must the system support?
* What is the skill level of each type of user? What type of training and documentation must be provided for each user?
* Do any users require special accommodations (large font size, ...)
* Must the system detect and prevent misuse? If so, what types of potential misuse must the system detect and prevent?
* ...>

**Documentation Requirements - Josh**

<Describe what documentation is required:

* on-line, printed, or both?
* what is the assumed skill level of the audience of each component of documentation?
* ...>

**Data Requirements- Chad**

* <Describe any data calculations: what formula will be used? to what degree of precision must the calculations be made? >
* <Describe any retained data requirements: exactly what must be retained? >
* < ... >

**Resource Requirements - MILES**

<Describe the system resources:

* skilled personnel required to build, use, and maintain the system?
* physical space, power, heating, air conditioning, ...?
* schedule?
* funding?
* hardware/software/tools?
* ...>

**Security Requirements - CHAD**

<Describe any security requirements:

* must access to the system or information be controlled?
* must one user's data be isolated from others?
* how will user programs be isolated from other programs and from the operating system?
* how often will the system be backed up?
* must the backup copies be stored at a different location?
* should precautions be taken against fire, water damage, theft, ...?
* what are the recovery requirements?
* ...>

**Quality Assurance Requirements - MILES**

<Describe quality attributes:

* What are the requirements for reliability, availability, maintainability, security, portability ...?
* How must these quality attributes be demonstrated?
* Must the system detect and isolate faults? If so, what types of faults?
* Is there a prescribed mean time between failures?
* Is there a prescribed time the system must be available?
* Is there a maximum time allowed for restarting the system after a failure?
* What are the requirements for resource usage and response times?
* ...>

**Supporting Material**

<Here is where you put all your analysis work from which you derived the above requirements. It may include UML or other diagrams, notes, memos, etc. >