**Requirements Specifications**

**InPress – Group 3**

Liu Wenbo (0970709)

Justin Kan (0843763)

Abdul Hadi Muhammad (0951862)

## 

Table of Contents

Change History 4

Project Drivers 5

THE PURPOSE OF THE PROJECT 5

The User Business or Background of the Project Effort 5

Goals of the project 5

THE CLIENT, THE CUSTOMER AND OTHER STAKEHOLDERS 5

The Client 5

The Customer 5

USERS OF THE PRODUCT 5

The Hands-On Users of the Product 5

Priorities Assigned to Users 5

User Participation 5

Maintenance Users and Service Technicians 5

Project Constraints 6

MANDATED CONSTRAINTS 6

Solution Constraints 6

Implementation Environment of the Current System 6

Partner or Collaborative Applications 6

Off-the-Shelf Software 6

Anticipated Workplace Environment 6

Schedule Constraints 6

Budget Constraints 6

NAMING CONVENTIONS AND DEFINITIONS 6

Definitions of all Terms 6

RELEVANT FACTS AND ASSUMPTIONS 7

Facts 7

Assumptions 7

Functional Requirements 7

THE SCOPE OF THE WORK 7

The Current Situation 7

The Context of the Work 7

Work Partitioning 8

THE SCOPE OF THE PRODUCT 11

Product Boundary 11

Product Use Case List 11

Individual Product Use Cases 12

FUNCTIONAL AND DATA REQUIREMENTS 16

Functional and Data Requirements 16

Nonfunctional Requirements 19

LOOK AND FEEL REQUIREMENTS 19

Appearance Requirements 19

Style Requirements 19

USABILITY AND HUMANITY REQUIREMENTS 19

Ease of Use Requirements 19

Personalization and Internationalization Requirements 19

Learning Requirements 19

Understandability and Politeness Requirements 19

Accessibility Requirements 19

PERFORMANCE REQUIREMENTS 19

Speed and Latency Requirements 19

Safety- Critical Requirements 19

Precision or Accuracy Requirements 20

Reliability and Availability Requirements 20

Robustness or Fault- Tolerance Requirements 20

Capacity Requirements 20

Scalability or Extensibility Requirements 20

Longevity Requirements 20

OPERATIONAL AND ENVIRONMENTAL REQUIREMENTS 20

Expected Physical Environment 20

Requirements for Interfacing with adjacent Systems 20

Productization Requirements 20

Release Requirements 20

MAINTAINABILITY AND SUPPORT REQUIREMENTS 20

Maintenance Requirements 20

Supportability Requirements 21

Adaptability Requirements 21

SECURITY REQUIREMENTS 21

Access Requirements 21

Integrity Requirements 21

Privacy Requirements 21

Audit Requirements 21

Immunity Requirements 21

CULTURAL AND POLITICAL REQUIREMENTS 21

Cultural Requirements 21

Political Requirements 21

LEGAL REQUIREMENTS 21

Compliance Requirements 21

Standards Requirements 21

Project Issues 22

OPEN ISSUES 22

OFF-THE-SHELF SOLUTIONS 22

NEW PROBLEMS 22

TASKS 22

MIGRATION TO THE NEW PRODUCT 22

RISKS 22

COSTS 22

USER DOCUMENTATION AND TRAINING 22

WAITING ROOM 22

IDEAS FOR SOLUTIONS 22

Table of Figures

Figure 1: Context of the Work 7

Figure 2: Work Partitioning 10

Figure 3: Product Use Case List 11

Figure 4: Create Instructor's Account Use Case 12

Figure 5: Create a Course Use Case 12

Figure 6: Create Questions Use Case 13

Figure 7: Post and UnPost an Assessment 13

Figure 8: Answer the Question Use Case 14

Figure 9: Check the Results Use Case 14

Figure 10: Question Data Analysis Use Case 15

Figure 11: Student Account Registration Use Case 15

Figure 12: Volere Cards 18

# Change History

1. The following table shows the change history for this test plan document.

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Version | 1. Date | 1. Author | 1. Comments |
| — | — | AHM, WL, JK | Original content. |
| 0 | October 10, 2013 | AHM, WL, JK | Initial check in |
| 1 | February 25, 2014 | AHM, WL, JK | Revision 0 Changes |
| 2 | April 11, 2014 | AHM, WL, JK | Revision 1 Changes |

Table 1: Change History

This document follows the Volere Requirements Specification Template. This template may be found at <http://www.volere.co.uk/pdf%20files/templateArchive.zip>.

# Project Drivers

## THE PURPOSE OF THE PROJECT

### The User Business or Background of the Project Effort

Design a web interface called “InPress”, which allows students in various universities to respond to real-time questions posted by their instructor using any internet-enabled device (i.e. iPads, iPhones, Android Phones/Tablets, PCs, Macs).

### Goals of the project

There are two goals in this project:

- Provide students with a free classroom interaction tool that will allow them to participate in classroom activities

- Allow students to learn more efficiently

## THE CLIENT, THE CUSTOMER AND OTHER STAKEHOLDERS

### The Client

Kevin Dunn, Chemical Engineering

### The Customer

Customers include colleges/universities, instructors and students.

## USERS OF THE PRODUCT

### The Hands-On Users of the Product

- Students and Instructors will be connecting to the web interface in order to interact with the various features.

- System Administrators will need to configure and setup the software environment through a web interface.

### Priorities Assigned to Users

- Key Users: Instructor and Students

- Secondary Users: System Administrators

### User Participation

- Students will answer questions posted by their instructor

- Instructors will be able to populate the system with questions

- Instructors will be able to analyze data received through student interaction

### Maintenance Users and Service Technicians

Maintenance users will be mainly System Administrators at the college/university, and Software Developers at InPress. System Administrators at the college/university will have to deal with any kind of operating discrepancy in the environment where the code is hosted.

Any other problems (i.e. a bug in the code) will first need diagnosing by the System Administrators, and if need-be a service ticket must be opened against the Development team at InPress.

# Project Constraints

## MANDATED CONSTRAINTS

### Solution Constraints

*Description*: InPress is a web-based system

*Rationale:* The InPress software will only be accessible via an Internet enabled device

*Fit Criterion*: Users need to use the Internet to communicate with InPress

### Implementation Environment of the Current System

The environment in which this system is going to be hosted is McMaster University. We will be hosting it on a departmental server (inpress.mcmaster.ca) in the Engineering Faculty. This software will be portable to other universities, but the software will have to be hosted internal in that university.

### Partner or Collaborative Applications

- Kevin Dunn, Chemical Engineering and Client

- Chemical Engineering System Administrators

- Dr. Rong Zheng, CAS Supervisor

- Dr. Spencer Smith, CAS Supervisor

### Off-the-Shelf Software

* **i-Clicker®:** Electronic devices that allows users to interact with the host through five buttons (A, B, C, D, E). The product supports multiple-choice only.
* Similar products already exist in the market (i.e. iClicker®). Most of the products only allow students to interact with the instructor, and do not have much functionality.

### Anticipated Workplace Environment

Colleges and Universities

### Schedule Constraints

All deliverables must be done within the deadline. The expected completed date is April 30/2014.

### Budget Constraints

N/A

## NAMING CONVENTIONS AND DEFINITIONS

### Definitions of all Terms

Students:People currently enrolled in an undergraduate or graduate program at a University/College

Instructor: An individual teaching an undergraduate/graduate course

System Administrator: Technical staff dealing with computers in the University/College

## RELEVANT FACTS AND ASSUMPTIONS

### Facts

- Current popular implementations (i.e. iClicker) allow students to answer multiple-choice questions only

- Current popular implementations (i.e. iClicker) require physical hardware

### Assumptions

- Hardware will be self-provided by the customer, and hardware failure will be addressed by on-site university IT personnel.

- We assume that the users have Internet access whenever they want to use the product

- Assume that students have some minimal knowledge (i.e. know the English language, know their Student No., etc.)

# Functional Requirements

## THE SCOPE OF THE WORK

### The Current Situation

- The software (web application) will require students and instructors to login, so that their responses will be saved

- The saved response can be used for data analysis (i.e. Number of Correct Responses)

- Students can use the web-application to respond to Instructor’s questions

- Easy setup for instructors by entering the questions before the lecture

- Students can review their answers, and the respective solutions after lecture

### The Context of the Work

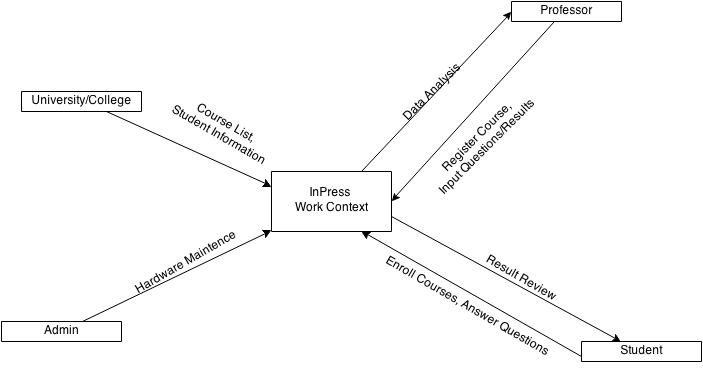


Figure 1: Context of the Work

### Work Partitioning

|  |  |  |
| --- | --- | --- |
| **Event Name** | **Input and Output** | **Summary** |
| Create an Instructor Account | Instructor’s Credentials (IN)  Created Account (OUT) | The system administrator will need to create an account for each instructor using InPress. They will need to provide InPress with basic information about the instructor, and will have the opportunity to create a unique username and password. |
| Login as an Instructor | Instructor’s Credential (IN)  Instructor’s Personalized Webpage (OUT) | Using the username and password provided by the System Administrator, instructors would be able to login and see their personalized page. |
| Login as a Student | Student’s Credential (IN)  Student’s Personalized Webpage (OUT) | Students will be able to login to InPress via their student number. Upon login, they will see their personalized webpage with courses they are enrolled in. |
| Create a Course | Instructor’s Credentials (IN)  Course Information (IN)  Created Course (OUT) | Courses can only be created through an instructor’s account. Once logged in, instructors may create a course by selecting the appropriate GUI button. Instructors will need to provide the web interface with basic course information (i.e. course name, course code, etc.). Once created, the course will appear on the instructor’s personalized webpage. |
| Delete a Course | Instructor’s Credentials (IN)  Course Information (IN) Course Deleted (OUT) | Upon selecting the appropriate GUI button on the instructor’s personalized webpage, instructors will be able to manage their courses. Through the “Remove a Course” UI, instructors will be able to delete a course. |
| Add a list of Questions/Solutions to a course | Instructor’s Credentials (IN)  Course Information (IN) List of Questions/Solutions (IN)  Course Populated with Questions/Solutions (OUT) | Upon selecting the appropriate GUI button on the instructor’s personalized webpage, instructors will be able to manage their courses. Instructors will be able to provide a list of questions/solutions to an assessment of a course. Once populated, instructors will be led back to their personalized webpage with the course now populated. |
| Post/Unpost an Assessment | Instructor’s Credentials (IN)  Course Information (IN) Posted Course (OUT) | Upon selecting the appropriate GUI button on the instructor’s personalized webpage, instructors will be able to manage their courses. Instructors will be able to post/Unpost an assessment for a course. |
| Effective Date (Assessment) | Instructor’s Credentials (IN)  Course Information (IN) Posted Course (OUT) | Upon selecting the appropriate GUI button on the instructor’s personalized webpage, instructors will be able to manage their assessments. All assessments have an effective date which restrict the assessment to be available only on that date. Instructors are able to change this date after an assessment has been created. |
| Data Analysis | Instructor/Student’s Credential (IN)  Course Information (IN)  Data Results (OUT) | Upon selecting the appropriate GUI button on the student/instructor’s personalized webpage, students will be navigated to the “Data Analysis” webpage. Through the  “Data Analysis” UI, users will be able to select a specific course, and would be able to analyze past questions and solutions for that course. |
| Add and Remove Students | Instructor’s Credentials (IN)  Course Information (IN) | Upon selecting the appropriate GUI button on the Instructor’s UI, Instructors will be able to add and remove students by adding/removing their student numbers. |
| LaTex in Questions/Solutions | Instructor’s Credentials (IN)  Course Information (IN)  LaTex Questions (OUT) | Upon selecting the appropriate GUI button on the Instructor’s UI, navigate to an assessment and add a question. Follow the instructions on how to add latex to your questions and/or solutions. Questions and Answers are outputted in LaTex. |

Figure 2: Work Partitioning

## THE SCOPE OF THE PRODUCT

### Product Boundary

In this case the work scope is the product boundary.

### Product Use Case List

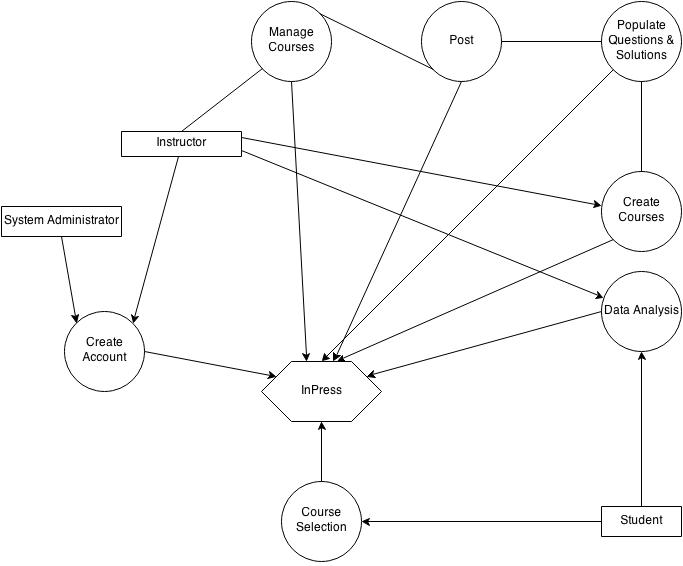


Figure 3: Product Use Case List

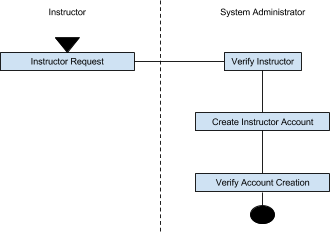
### 

### Individual Product Use Cases

1. **Product Use Case Name:** Create Instructor’s Account

**Trigger:** System Administrator Manage Account Page

**Preconditions:** Login as System Administrator

**Interested Stakeholders:** The client, customer

**Actor:** Web browser

**Activity Diagram:**

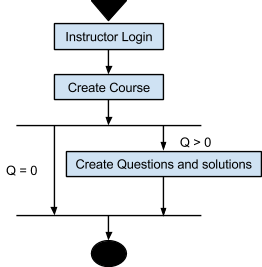
Figure 4: Create Instructor's Account Use Case

1. **Product Use Case Name:** Create a Course

**Trigger:** “Create a Course” on Manage Courses webpage

**Preconditions:** Have to login as an instructor

**Interested Stakeholders:** The client, customer

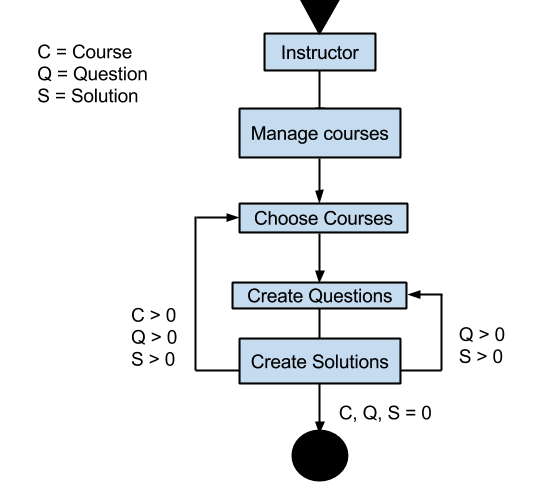
**Actor:** Web browser

**Activity Diagram:**

Figure 5: Create a Course Use Case

1. **Product Use Case Name:** Create Questions

**Trigger: “**Create Questions” UI Button in the Course’s Personalized Webpage

**Preconditions:** Have to login as an instructor

**Interested Stakeholders:** The client, customer

**Actor:** Web browser

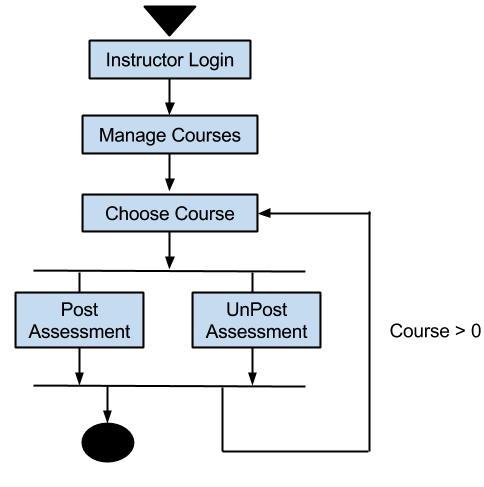
**Activity Diagram:**

Figure 6: Create Questions Use Case

1. **Product Use Case Name:** Post/Unpost an Assessment

**Trigger:** “Post/Unpost” UI Button in the Course’s Personalized Webpage

**Preconditions:** Have to login as an instructor

**Interested Stakeholders:** The client, customer

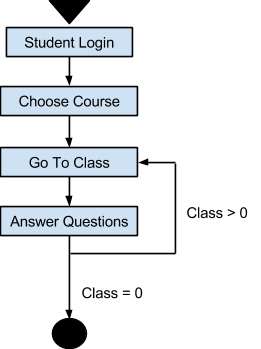
**Actor:** Web browser

**Activity Diagram:**

Figure 7: Post and UnPost an Assessment

1. **Product Use Case Name:** Answer the question

**Trigger:** “Go To Class” UI button in the Course’s personalized webpage

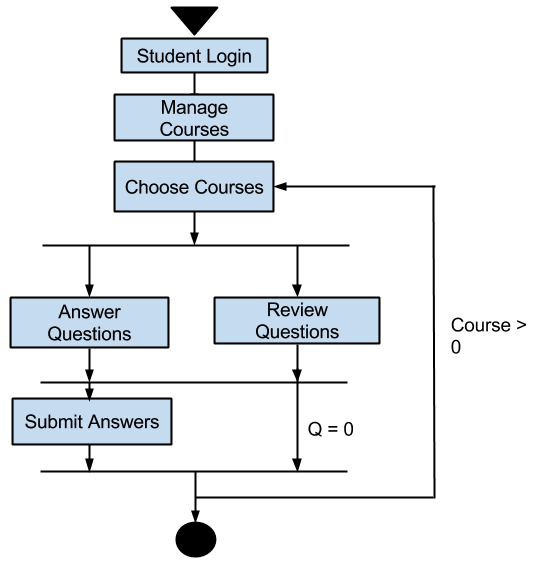
**Preconditions:** Have to login as a Student

**Interested Stakeholders:** The client, customer

**Actor:** Web browser

**Activity Diagram:**

Figure 8: Answer the Question Use Case



1. **Product Use Case Name:** Check the results

**Trigger:** “Results” UI Button on the Course’s Personalized webpage

**Preconditions:** Login as an instructor/student, must be enrolled in the course

**Interested Stakeholders:** The client, customer

**Actor:** Web browser

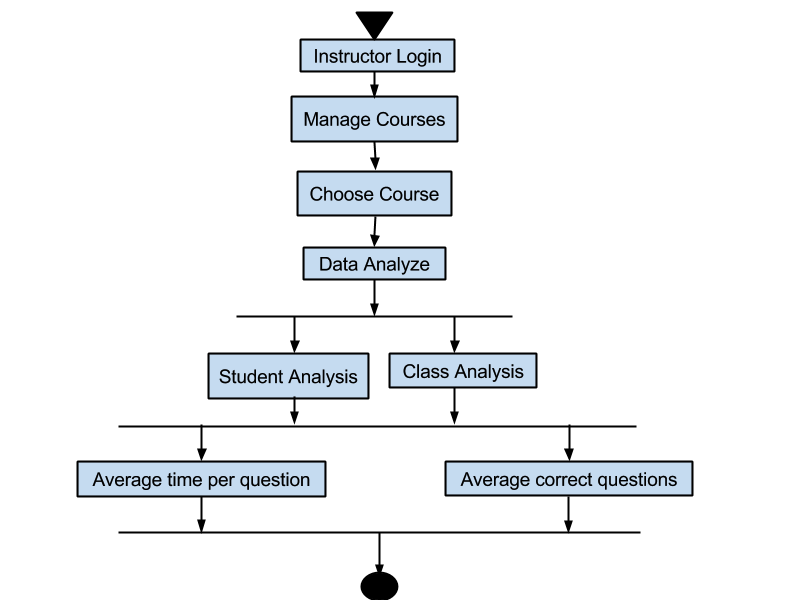
**Activity Diagram:**

Figure 9: Check the Results Use Case

1. **Product Use Case Name:** Question Data Analysis

**Trigger:** “Data Analysis” UI Button in the Course’s personalized webpage

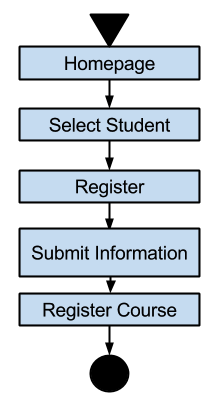
**Preconditions:** Have to login as Instructor and posted this question

**Interested Stakeholders:** The client, customer

**Actor:** Web browser

**Activity Diagram:**

Figure 10: Question Data Analysis Use Case

****

1. **Product Use Case Name:** Student Account Registration

**Trigger:** Home page “Register” button

**Preconditions:**

**Interested Stakeholders:** The client, customer

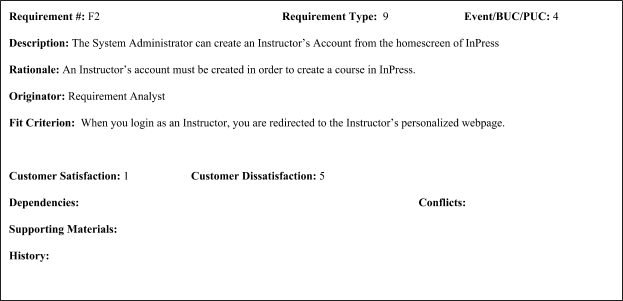
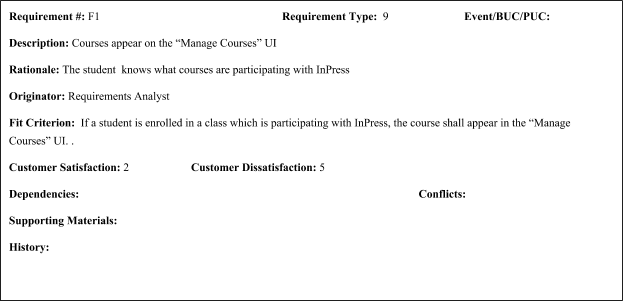
**Actor:** Web browser

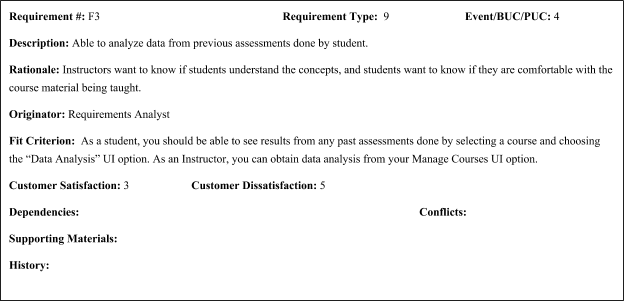
**Activity Diagram:**

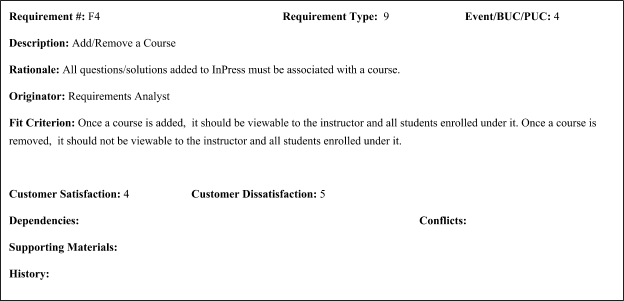
Figure 11: Student Account Registration Use Case

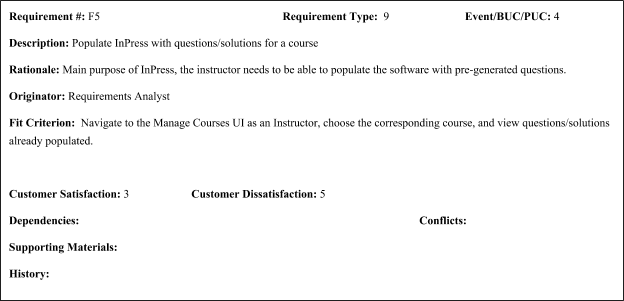
## FUNCTIONAL AND DATA REQUIREMENTS

### Functional and Data Requirements









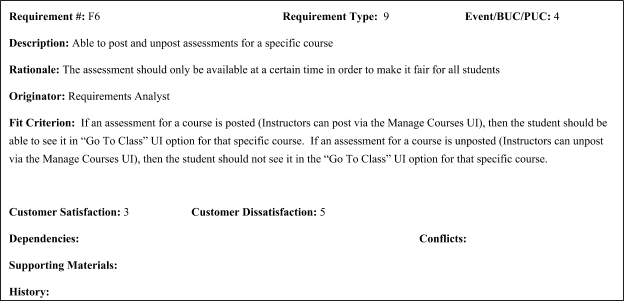


Figure 12: Volere Cards

# Nonfunctional Requirements

## LOOK AND FEEL REQUIREMENTS

### Appearance Requirements

The User Interface for the web-application will be a simple interface with very few buttons.

### Style Requirements

The style will be user friendly, keeping button clicks to a minimal in order to achieve various tasks.

## USABILITY AND HUMANITY REQUIREMENTS

### Ease of Use Requirements

The web-application should be easy to use for every type of users out there, from novice to power-users.

### Personalization and Internationalization Requirements

- Since each student and instructor have different courses, they will have their own webpages

- The web-application will be in English.

### Learning Requirements

- The learning curve will be kept minimal. It should not take them more than five minutes to understand how to use this web-application.

- For the System Administrator or Technicians, it should not be difficult to install the application, maintain and create Instructor ID’s.

### Understandability and Politeness Requirements

- Any type of user can use the web application easily

- Hide functions that will confuse users

- There should be no grammar mistakes and inappropriate language anywhere on the web-application.

### Accessibility Requirements

- For Mobile Apps, it will be on Apple’s App Store, or Google’s Play store

- For Laptops, there will be a link that can access it directly (something similar to Avenue).

## PERFORMANCE REQUIREMENTS

### Speed and Latency Requirements

- Communicating with the database should be fast, to reduce latency.

- Internet Connection between the customer and the hosts should be fast if it is on campus.

### Safety- Critical Requirements

- Student data should not be accessible by any other student

- Instructors data should only be accessed by the instructor themself or the system administrator.

- Only the System Administrator has access to the administration site.

### Precision or Accuracy Requirements

- There should be no errors when saving the students answers, even after they changed their answers.

- The web application should not be retrieving incorrect student information.

- Students, administrator, and instructors should have their own unique ID’s and should not be mixed up

### Reliability and Availability Requirements

- The system will be reliable because it will be running on a server within the institute’s data center.

- Since it is inside a data center, it will be available 24 hours a day 7 days a week.

### Robustness or Fault- Tolerance Requirements

- The web-application will be robust enough to recover from errors (such as duplicated login)

- Zero fault-tolerance. This is because if the students’ data are saved incorrectly, data analysis will not be accurate. Hence, defeating the point of the data analysis.

### Capacity Requirements

- The application should be scalable, meaning the max capacity will depend on the hardware (system and storages)

### Scalability or Extensibility Requirements

- Infrastructure similar to cloud. An increase of storage space, or an upgrade for the system (CPU / Memory) will be sufficient.

### Longevity Requirements

- It should last until it is no longer used. There will be maintenance patches, and the robustness should recover from failures.

## OPERATIONAL AND ENVIRONMENTAL REQUIREMENTS

### Expected Physical Environment

- It could be anywhere, from classroom to home.

- The hosting server will be in the school’s data center

### Requirements for Interfacing with adjacent Systems

Not Applicable

### Productization Requirements

Not Applicable

### Release Requirements

This product will be opened sourced and will have regular maintenance.

## MAINTAINABILITY AND SUPPORT REQUIREMENTS

### Maintenance Requirements

- Periodic patches and maintenance to ensure the system is up-to-date and bug-free

### Supportability Requirements

- Backup should be made every month for current courses.

### Adaptability Requirements

- The web-application should be able to adapt into any Linux environment

- Migrating into a new box should be very easy

## SECURITY REQUIREMENTS

### Access Requirements

- Only the System Administrator can make changes to the web-application.

- Instructors can only see their own course data along with their enrolled students’ data

- Students should only access their only account

### Integrity Requirements

- Security algorithms will be in place to prevent data corruption

- Minimum of RAID 5 array will be in place to recover from hard disk failures.

### Privacy Requirements

- Only the System Administrator may see everyone’s data

- The instructors will only see their own course data along with their enrolled students' data.

- Students can only see their own data.

- The main implementation will be hidden from everyone.

### Audit Requirements

- The web-application will follow the institute’s data center security compliances.

### Immunity Requirements

Not Applicable as it will depend on the Institute’s Security.

## CULTURAL AND POLITICAL REQUIREMENTS

### Cultural Requirements

Not applicable

### Political Requirements

Not applicable

## LEGAL REQUIREMENTS

### Compliance Requirements

- Web-application is open-sourced, and will follow the relevant licenses.

- Web-application should not involve any patent wars.

### Standards Requirements

**-** Everything is confidential, and should not be released to anyone except authorized individuals

- Will not be affiliate with any Patents.

# Project Issues

## OPEN ISSUES

* We do not know where the server is physically located
* We do not know the specifications of the server

## OFF-THE-SHELF SOLUTIONS

N/A

## NEW PROBLEMS

N/A

## TASKS

* 1. Gather missing requirements
  2. Develop prototype
  3. Get feedback about prototype
  4. Provide arguments why current prototype is ideal by comparing it to current of-the-shelf products.

## MIGRATION TO THE NEW PRODUCT

Not applicable.

## RISKS

There is a risk of being unable to migrate the web-application to a new server if the current server cannot support any more resources (i.e. increase of CPU or memory). Although everything can be made into an image and be migrated to a new server, there is no promise that storage will not have errors.

## COSTS

Not in the requirement

## USER DOCUMENTATION AND TRAINING

There will be documentation for all three groups of people: the administrators (including system administrators / technicians), instructors and students. This documentation is crucial for the administrators because they will be the ones that have to set up the system.

## WAITING ROOM

Not applicable

## IDEAS FOR SOLUTIONS

Not applicable

Not applicable