Automated System for U-Build Simulation Project

Final Project Report

04/24/2018

Guided By: Prof. Bengisu Tulu

Prepared By:

Anuj Vyas
Manisha Chouhan
Nevetha Ramesh
Suraj Patil
Vishakha Jadhav

Contents

Abstract

This project has been initiated as part of the course MIS 573- Systems Design and Development. Through this project, we are developing a system for improving and automating existing process of assignment evaluation for U-Build simulation Project at Worcester Polytechnic Institute. We have used several techniques required for planning, analysis, design and implementation phases of software development life cycle for this project. Also, we have studied and utilized the necessary infrastructure, required resources, and other essential aspects. This report covers all the required aspects for development of 'Automated System for U-Build simulation Project'.

Executive Summary

Automated System for U-Build simulation project offers a platform to generate financial statements based on the data provided by the students and to improve the evaluation process of the submissions.
Financial statement is a collection of records of financial activities, position, results and cash flow of a business, person or other entity.

The system ensures that the learning value for students is enhanced by process automation and also saves time and effort for the evaluator by automatically comparing the students' submission results and the actual results.

Technical Feasibility:

This system is technically feasible, although there are a few risks involved.

- Risk regarding familiarity with the application:
- The new system, from the users' point of view, will be similar for the process being followed however, the platform is going to be different.
- Risk regarding familiarity with the technology:
- The system will be developed for users who might not have complete understanding and expertise of the technology.
- The software used to develop the application is newly introduced to the team who have a beginner's level of expertise for the same.

Economic Feasibility:

Conservative estimates were drawn from the cost-benefit analysis which depict the system has a reasonable chance of significantly enhancing the current process.

ROI over 3 years: 27.92 %
Break-even occurs after 2.7 years

Intangible costs and benefits:
Enhanced process efficiency.
Enhanced users' experience.

Organizational Feasibility:

From an organizational perspective, this project has low risk. The project sponsor and the future users of the application have a keen interest in developing this application for reducing the overall effort and increase the learning value for the students.

The team believes that the users of the system, i.e., students and evaluators will be appreciating this automation as it will accelerate the process of uploading the data and generating financial statement and evaluating the students' assignments.

Also, the developed application can be accessed by faculty members within WPI who use the same simulation project as assignments in their courses. The to-be system will be used by the students to submit their financial statements and will be used by the Professors and Teaching Assistants, i.e., evaluators, for evaluating them.

Additional Comments:

- We will need training sessions for the users of the system to organize at the begging of the course.
- We will need an administrator if the users require some significant changes.

Process to be followed by the proposed system

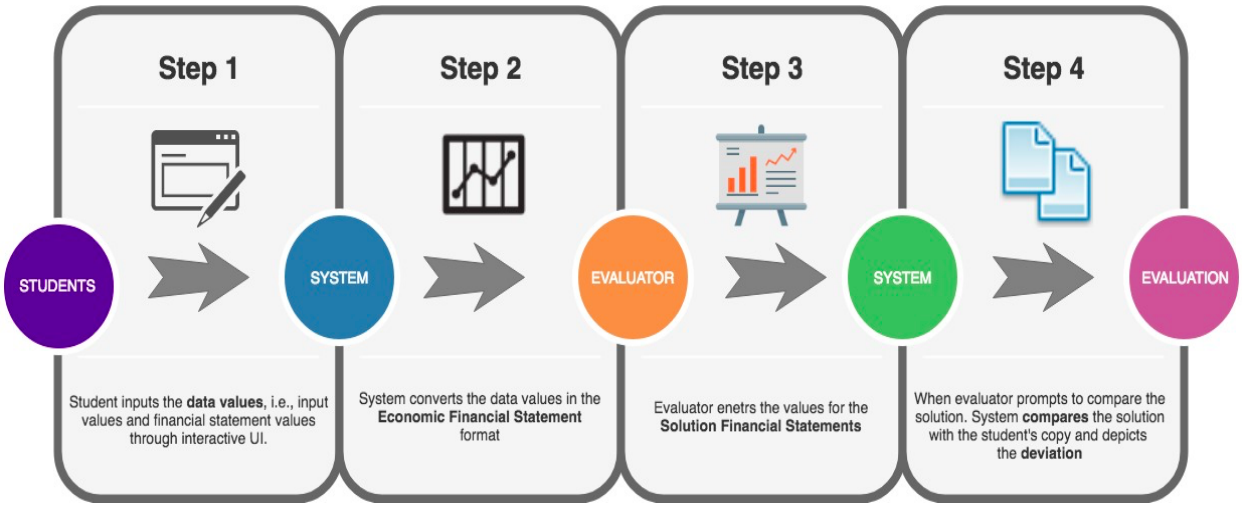


Figure 1: Proposed System Process

System Architecture Design

The platform being used for the development is Mendix. The system architecture followed by mendix is- Component Based Architecture which provides flexibility and scalability with less complexity for the application development.

Mendix provides below architecture options:

- Minimal Server Architecture
- Hosting with separate Database Server
- Separate Database Server and separate Web Server
- Separate Mendix Web Server in a DMZ

The option selected for U-Build automation process is 'Minimal Server Architecture' which is depicted in the below diagram.

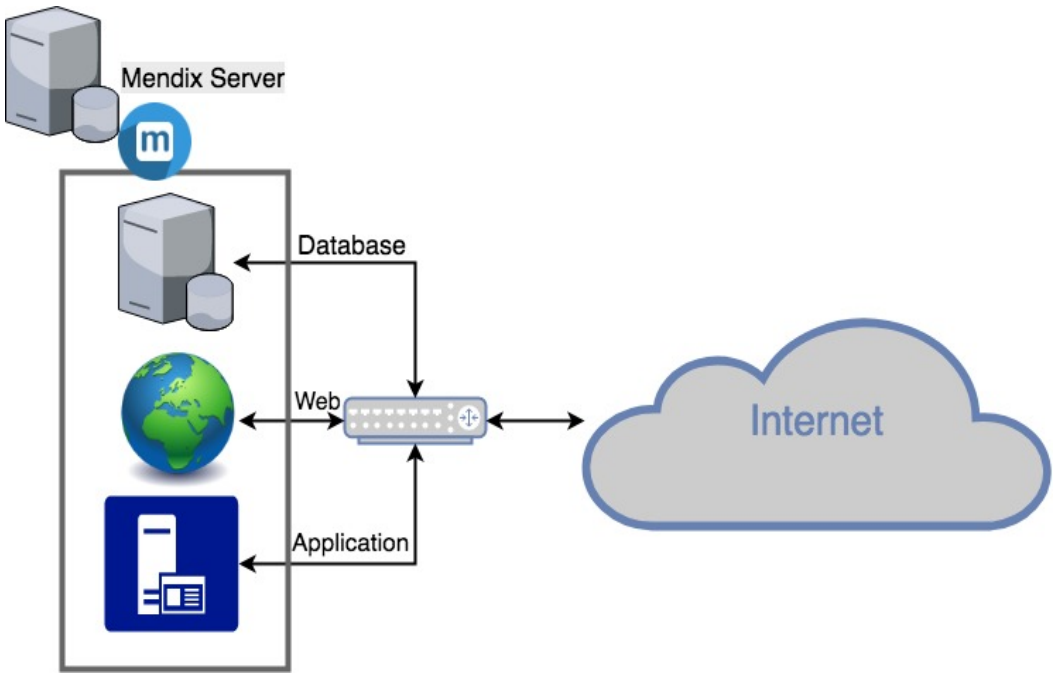


Figure 2: Minimal Server Architecture

Details:

Firewall Proxy Server:

This is an optional server provided in front of the Mendix Web Server for network security.

Web/ Application/ Database Server:

- It utilizes the database server hosted on the same server instance.
- It hosts the complete application.
- All the communication between the client and the server is done via web service or a web proxy.

File Server:
Optional server for the storage of the uploaded or generated files. This is accessed by the application server.

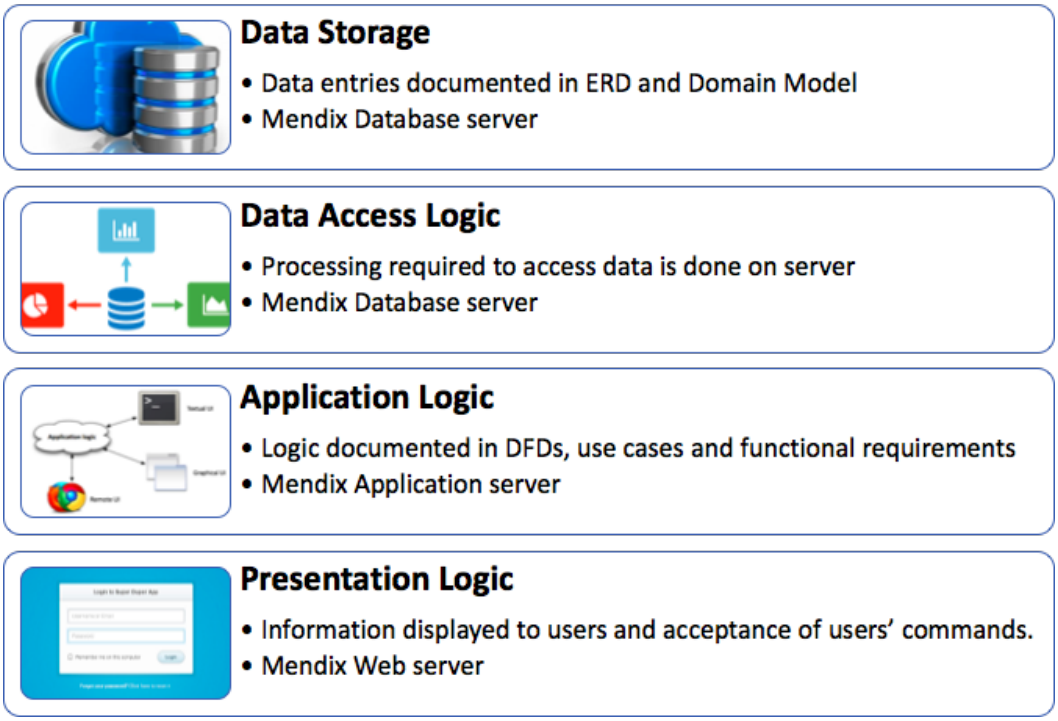


Figure 3: Basic functions of system architecture

The architecture design as depicted in Figure <> has four basic functions that have been implemented as a part of the U-Build system.

Data Storage:
Application related data is currently being stored on the Mendix database server. In future, if required, WPI Database can be used for Data Storage to integrate the student details directly.
Data that will be stored in the form of persistable and non-persistable entities are documented in the Domain Model.

Data Access Logic:
Data access logic refers to the processing required to access data stored in the database. It is implemented using the database server.
X-path and Associations are the data access logic used.

Application Logic:
The basic logic of the application is documented in the data flow diagrams, use cases, and functional requirements. The application logic currently resides on the application server.

Presentation Logic:
Through the presentation logic, application displays required information to the user and accept user's commands through the user interface. The presentation logic is stored on the web server.

Hardware components required for the proper functioning of the application are as follows:

- Client Computers: Client computers are the input and output devices employed by the users of the system and are desktop or laptop computers.
- Server: Mendix server
- Network: The network connects the Client Computers to the server. The users can use any available network source to run the application on the web. The network that connects these computers can vary in speed which can have a significant impact on the performance of the system.

Reasons for selecting the above-mentioned architecture are as follows:

- For the initial implementation of the project, connectivity to the in-house database server, i.e., WPI's database server, is not required.
- Using Mendix as the development platform, we can ensure that all the related files are stored in the secured system server.
- The web-based system will allow users to use the system anytime, anywhere.
- The system does all the calculations for comparison of the financial statement values at the back-end keeping the user interface simple.

Domain Model

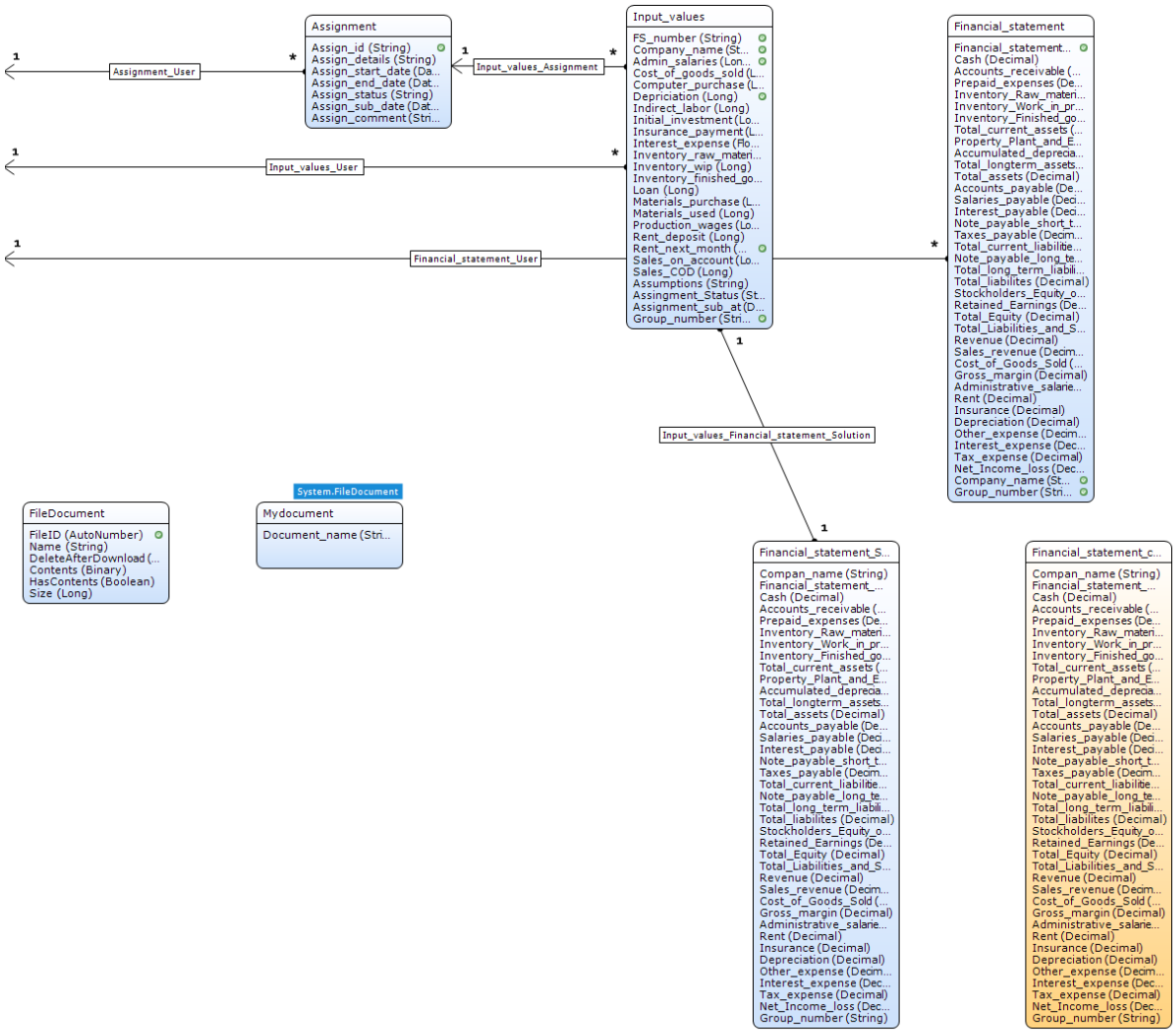


Figure 4: Domain Model for the final system

User Stories

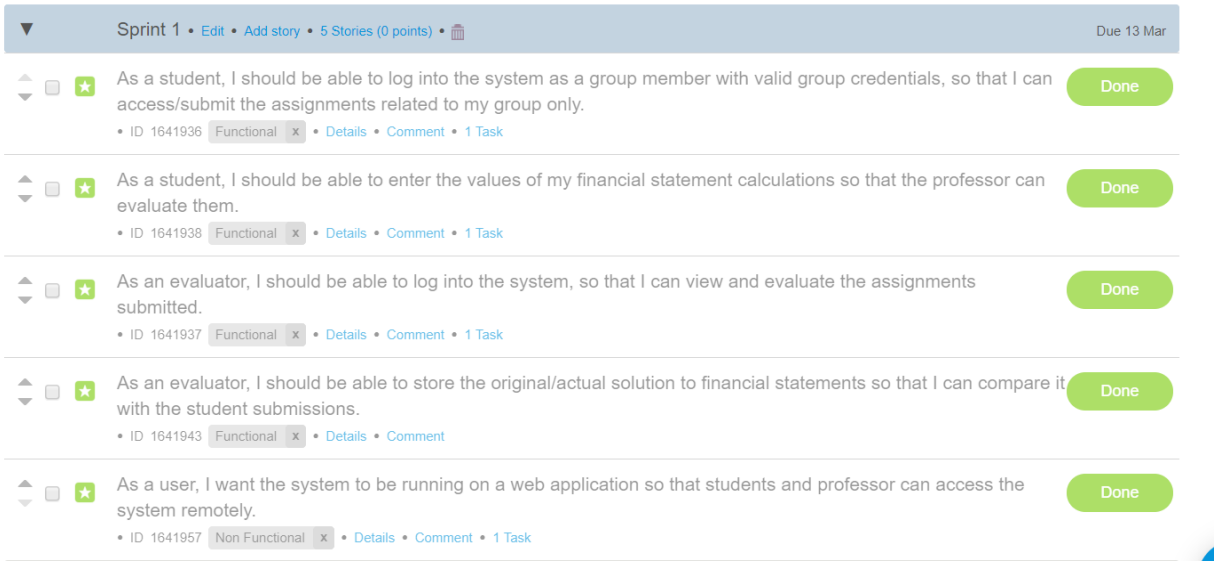


Figure 5: User stories for Sprint 1

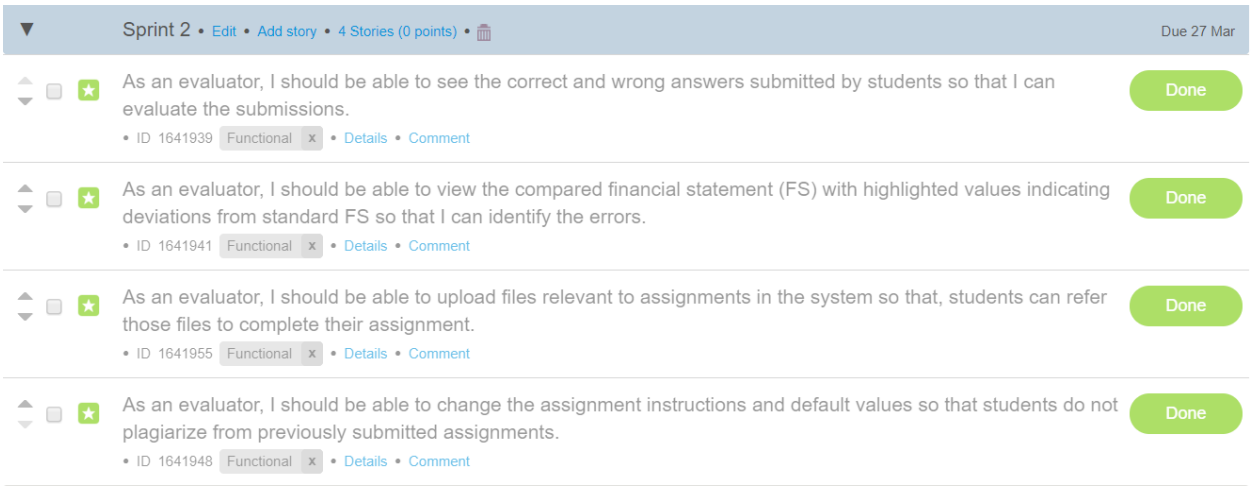


Figure 6: User stories for Sprint 2

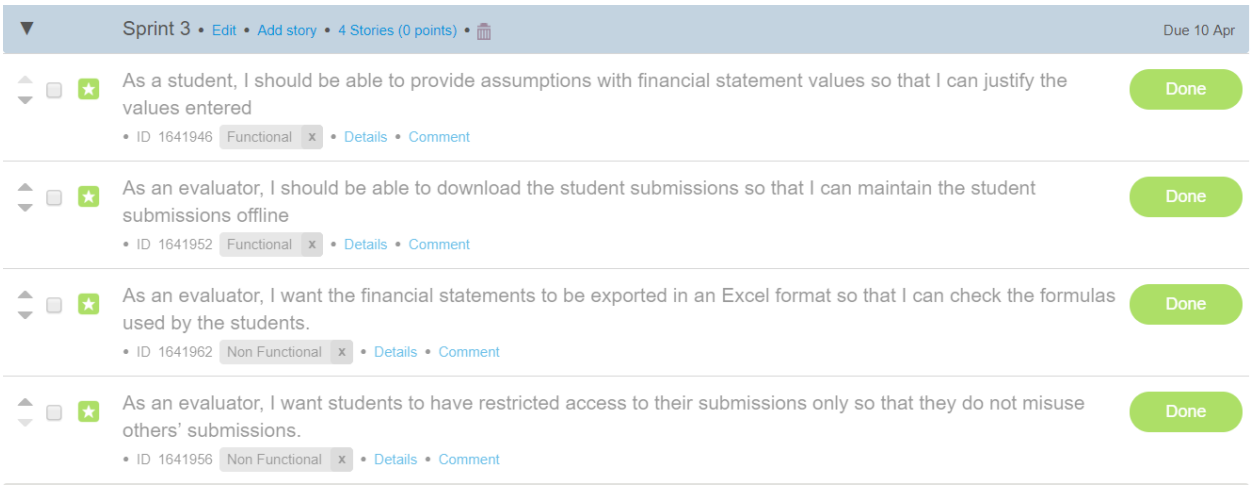


Figure 7: User stories for Sprint 3

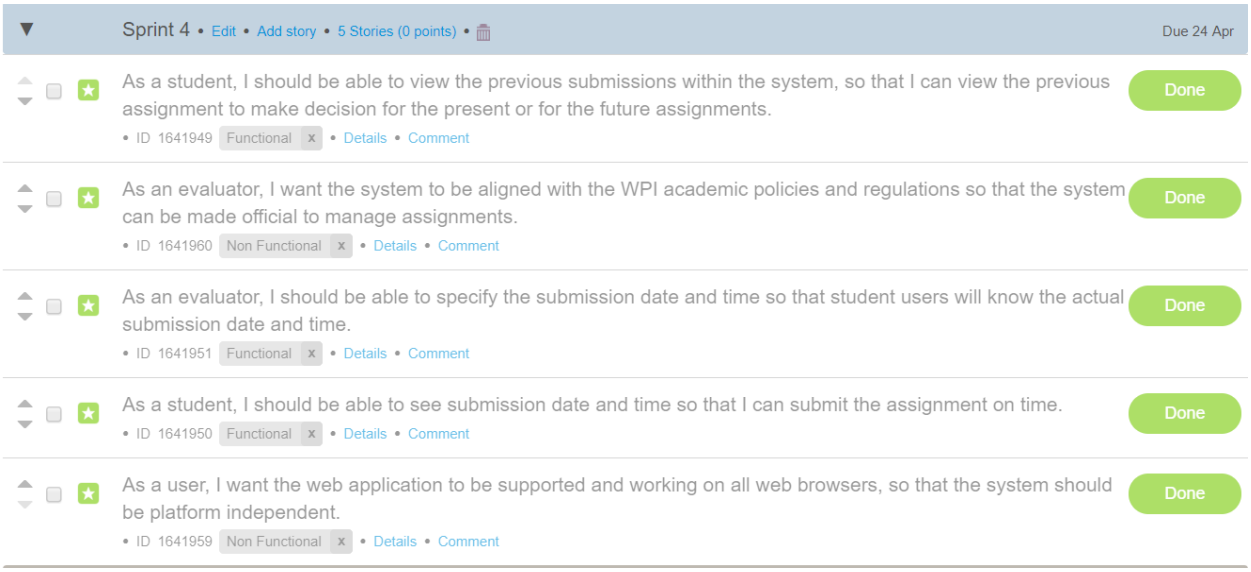


Figure 8: User stories for Sprint 4

Use Scenarios

- Administrator user: Creates user account with credentials for specific roles.

Use scenario: Creates user account with credentials for specific role
1. Administrator will create a new user account 2. Administrator will enter User name, User role and New password for the account. 3. Admin will save or cancel the changes he/she makes. 4. Admin will update the user account to make any changes if required. 5. Admin will delete the user account if wrongly created or if no more needed.

- Student user: Submits an assignment

Use scenario: The Hurry-up Student
1. Student will download the specific assignment instructions. 2. Student will enter the input and financial statement values. 3. Student will confirm the submissions of the values. 4. Student can view the submission status based on submission time.

Use scenario: The Browsing Student
1. Student may download the specific assignment instruction or search for the assignment instructions using file name. 2. Student may view previous financial submissions made and may review them. 3. Student may submit the assignment and continue to on the same page to view the submission in the financial statements layout. 4. Student may review the assignment submitted and decide on editing/deleting the values that they had provided. 5. Student may re-submit the assignment with correct values. 6. Student can view the submission status based on submission time.

- Evaluator: Evaluator evaluates the Group's assignment

Use scenario: The Hurry-up Evaluator
1. Evaluator views the input values and financial statement submissions made by the groups. 2. Evaluator will view the comparison against the expected solutions.

Use scenario: The Browsing Evaluator
1. Evaluator views the submission made by groups 2. Evaluator may view the comparison against the expected Financial statement solution. 3. Evaluator may upload/modify assignment files. 4. Evaluator may add/modify the assignment start and deadline dates and time.

Test Plan

Non-trivial integration tests conducted are included below with the test results. We have performed several dry runs and two formal test runs for these test cases. Apart from these integration tests we have performed through Unit, System and User Acceptance tests.

Unit Testing:

- Black box testing is performed.
- Screen wise testing of all elements and data fields.
- Field data validations.
- Calculated fields are validated against actual formulas.
- All the interfaces for all the user roles are thoroughly tested.
- Small functionalities or button operations were verified.

Integration Testing:

- Integration test cases were designed for majority of interface functionalities.
- Role or login specific data availability is verified.
- Verification of data referenced by different users from the same source.
- Exchange of data is verified for all the upload and download functionalities.

System Testing:

- All the functionalities are tested against the requirements.
- Generic usability aspect is tested and enhanced accordingly.
- User manual is verified.

User Accepted Testing:

- Alpha testing is performed.

RUN 1

Test Plan

Tester: Nevetha Ramesh **Date designed:** 04/09/2018 **Date conducted:**04/09/2018

Result: ✓ **Passed** **Open Items**

Test ID: INT1 **Requirement addressed:** Login functionality

Objective: User should be able to login successfully, only with valid login and password credentials created by Admin.

Test cases: **Interface**	**Data Field**	**Value Entered**
1. Login Screen	Username, Password	Group1, Group1
2. Login Screen	Username, Password	Group1, Apple
3. Login Screen	Username, Password	Group1,(Blank)
4. Login Screen	Username, Password	Evaluator1, Evaluator123
5. Login Screen	Username, Password	Evaluator1, Pineapple
6. Login Screen	Username, Password	Evaluator1, (Blank)

Script

1. Create valid user credentials as user- Admin.
2. Log-out from the application.
3. Log-in into the application with newly created credentials.

Expected results/notes

Tests 1,4 are valid User name, Password. All others should be rejected.

Actual results/notes

Tests 1,4 are accepted. Tests 2,3,5 and 6 were rejected with correct error message.

Test Plan

Tester: Nevetha Ramesh **Date designed:** 04/09/2018 **Date conducted:**04/09/2018

Result: Passed ✓ **Open Items**

Test ID: INT2 **Requirement addressed:** Access and download files

Objective: Users should be able to download the files to which they have access to in the required layout and PDF format.

Test cases:		
Interface	**Data Field**	**Value Entered**
1. Student - Submit Assignment	Submit Assignment	Input and Financial statement values
2. Evaluator - Assignment Information Files	New (File Manager)	New or modified uploaded instruction file

Script

1. Login using the valid student credentials
2. Click on 'Submit Assignment'
3. Enter the required data values
4. Click on 'Submit Assignment' button
5. Navigate to the 'Previous submission'
6. Click on 'Download financial statement' button
7. Logout from the application
8. Login using valid evaluator credentials
9. Follow the navigation- Student Submission →Financial Statement →View Comparison
10. Click on 'Download Financial Statements' button

Expected results/notes:

- Submitted assignment should be available for download for the evaluator and to that student group only in PDF format in the required template.
- Instruction files uploaded by evaluator should be downloadable to all student groups.

Actual results/notes

- Assignment submitted by Group 1 was available for download only for the evaluator and Group 1, but the file is different from the required template and not available as PDF but as Excel.
- Assignment Instruction files uploaded by evaluator was available for download to all the student groups.

Bug #1 : The downloaded financial statements are not in the specified financial statement format, which is not as expected.

Test Plan

Tester: Nevetha Ramesh **Date designed:**04/10/2018 **Date conducted:**04/10/2018

Result: ✓ **Passed** **Open Items**

Test ID: INT3 **Requirement addressed:** Assignment information visibility

Objective: Assignment information created by evaluator should be available for viewing for all student groups.

Test cases:		
Interface	**Data Field**	**Value Entered**
1.Evaluator - Assignment Deadline Information	Month, Assignment start date, Assignment end date (Assignment Information)	1, 4/21/2018, 12.00 AM, 4/26/2018, 12.00 AM

Script

1. Login as Evaluator.
2. Select 'Assignment Deadline Information'
3. Select 'New' and Enter the above given values and Log-out
4. Log-in as student
5. On the Dashboard page, check the Assignment dates

Expected results/notes

The start and end date/time for the month entered by evaluator should be visible for all student groups in their home page.

Actual results/notes

All student groups were able to view the assignment information entered by the evaluator according to the month specified.

Test Plan

Tester: Nevetha Ramesh **Date designed:**04/11/2018 **Date conducted:**04/11/2018

Result: ✓ **Passed** **Open Items**

Test ID: INT4 **Requirement addressed:** Submission status

Objective: Student should be able to see the on-time or late submission status after making a submission.

Test cases: **Interface**	**Data Field**	**Value Entered**
1.Student - Submit Assignment	Input values and Financial Statement values - Group Number, Month, company name	1, 1, abc

Script

1. Follow the script in INT3
2. Login as student and select "Submit Assignment"
3. Enter the above values and confirm submission
4. Navigate to student dashboard page
5. Check the status and compare it with Assignment deadline and submission time

Expected results/notes:

- On-time or Late submission status should be displayed for all submission made by that group.
- Status should be on-time if submission time is less than equal to deadline time
- Status should be late submission if submission time is greater than deadline time

Test Plan

Tester: Nevetha Ramesh **Date designed: 4/12/2018 Date conducted: 4/13/2018**

Result: Passed ✓ **Open Items**

Test ID:INT5 **Requirement addressed:** Financial statement Comparison results

Objective: The financial statement values of students should be compared with the expected financial statement generated using the input values of students and deviations should be specified.

Test cases:		
Interface	**Data Field**	**Value Entered**
Student - Submit Assignment	Input values and Financial Statement values - Group Number, Month, company name, Rent	1,1, abc, 1000
Evaluator - Student submission	Financial Statement - Student Value, Differences	Financial Statement 1 - Selection

Script

1. Login as student and select "Submit Assignment"
2. Enter the values for input and financial statement values starting from month 1 as given in the Excel reference document by the sponsor and confirm submission
3. Log-out from student account
4. Log-in as an Evaluator and select "Student Submission"
5. Select "Financial Statement" and select the financial statement tabs starting from month 1
6. Select "view comparison" and verify the comparison values with that of the Excel reference document

Expected results/notes

- The comparison between input values and financial statement generated values should match with that of the reference document provided by sponsor.
- Comparison should be made using the Financial statement values and generated financial statement from the input values.

Actual results/notes:

The comparison values were different from that present in the reference document. The compared values were not obtained from the student submissions but it considered default values present on the evaluator side.

RUN 2

Test Plan		
Tester: Nevetha Ramesh	**Date designed:** 4/10/2018	**Date conducted:**4/10/2018
Result: ✓ **Passed**	**Open Items**	

Test ID: INT2 **Requirement addressed:** Access and download files

Objective: Users should be able to download the files to which they have access to in the required layout and PDF format.

Test cases:		
Interface 3. Student - Submit Assignment 4. Evaluator - Assignment Information Files	**Data Field** Submit Assignment New (File Manager)	**Value Entered** Input and Financial statement values New or modified uploaded instruction file

Script

1.Login using the valid student credentials
2.Click on 'Submit Assignment'
3. Enter the required data values
4. Click on 'Submit Assignment' button
5. Navigate to the 'Previous submission'
6. Click on 'Download financial statement' button
7. Logout from the application
8. Login using valid evaluator credentials
9. Follow the navigation- Student Submission → Financial Statement → View Comparison
10. Click on 'Download Financial Statements' button

Expected results/notes: ● Submitted assignment should be available for download for the evaluator and to that student group only in PDF format. ● Instruction files uploaded by evaluator should be downloadable to all student groups.
Actual results/notes ● Assignment submitted by Group 1 was available for download only for the evaluator and Group 1 in the same layout of a financial statement in PDF format. ● Assignment Instruction files uploaded by evaluator was available for download to all the student groups.

Test Plan

Tester: Nevetha Ramesh **Date designed: 4/12/2018 Date conducted: 4/13/2018**

Result: ✓ **Passed** **Open Items**

Test ID: INT5 **Requirement addressed:** Financial statement Comparison results

Objective: The financial statement values of students should be compared with the expected financial statement generated using the input values of students and deviations should be specified.

Test cases:		
Interface	**Data Field**	**Value Entered**
Student - Submit Assignment	Input values and Financial Statement values - Group Number, Month, company name, Rent	1,1, abc, 1000
Evaluator - Student submission	Financial Statement - Student Value, Differences	Financial Statement 1 - Selection

Script

1.Login as student and select "Submit Assignment"
2.Enter the values for input and financial statement values starting from month 1 as given in the Excel reference document by the sponsor and confirm submission
3.Log-out from student account
4. Log-in as an Evaluator and select "Student Submission"
5. Select "Financial Statement" and select the financial statement tabs starting from month 1
6. Select "view comparison" and verify the comparison values with that of the Excel reference document

Expected results/notes

- The comparison between input values and financial statement generated values should match with that of the reference document provided by sponsor.
- Comparison should be made using the Financial statement values and generated financial statement from the input values.

Actual results/notes:

The compared values are matching to that present in reference document.
Comparison result is shown only from the student given values and not by considering the evaluator side default values.

Test Plan		
Tester: Nevetha Ramesh **Date designed:** 4/15/2018 **Date conducted:**4/16/2018 **Result:** ✓ **Passed** **Open Items**		
Test ID: INT6 **Requirement addressed:** Group access functionality **Objective:** Submissions made by one group should not be accessible to any other group for viewing or downloading.		
Test cases: **Interface**	**Data Field**	**Value Entered**
Script 1.Log-in using valid student credentials for Group 1 2. Follow the steps in INT2 from step 2 to step 5 3. Log-out of the application 4. Log-in as Group 2 using valid credentials 5. Navigate to "Previous submission" screen 6.Check if the submission made by Group 1 is available either for viewing or downloading. 7. Log-out from the application.		
Expected results/notes: • Submitted assignment should be available for viewing only to the students belonging to Group 1 • A student logged in with Group 2 credentials should not be able to view submission made by Group 1		
Actual results/notes • Assignment submitted by Group 1 is available for viewing only to the students of Group 1 • A student logged in with Group 2 credentials is not able to view submission made by Group 1		

Conversion Strategy

Based on the understanding of users and working process of current system and new system, the following conversion strategy seems to be a viable option.

- **Conversion Style: Parallel**

Parallel style of conversion is chosen where users can continue to use the existing system with new system in parallel so that they can see the effectiveness of the new system and realize the simplicity in it. This will also help them to migrate to the new system in a seamless way by getting good hands on it.

- **Conversion Location: Simultaneous**

As the users and system usage is not spread across multiple locations, there is no necessity to launch the system at one or few locations to study the performance. So, it will be launched in one single go for the location where it is used now.

- **Conversion Modules: Whole-system**

As the number of distinct users are limited and some of the user scenarios are dependent on others, to launch the whole system is the best option as compared to modular approach.

Advancements from Prototype

The system has been modified and improved based on the constructive feedback received from the project sponsor on the prototype developed. Following are the changes that has been incorporated into the prototype to develop a final version of the system.

- Extended the generation of financial statements to all 6 installments in which students may submit.
- Standardized the theme and aesthetics across the entire system.
- Modified the layout of the screens to align with format of financial statement.
- Included validation rules and comments wherever required.
- Created submission status according to the assignment deadline.
- Included navigation panels and more navigation features.
- Enhanced the access authorizations to view files and submissions.

- Modified the UI design of screens to improve the experience.
- Included sign-out options for users.
- Removed few redundant functionalities.

Organizational Change Issues

The new system that has been developed will not replace the existing system (which is Canvas) where the students submit their financial reports and evaluator manually checks and evaluates them. The new system would be a supplement to the existing one which will make it easier for the evaluator to get rid of the manual checking process.

- The **project sponsor** is also one of the users of the system who wanted this change in first place which reduces the chances of resistance from one side.
- The **change agent** will be out team who plan and implement the change while remaining out of the business process.
- The **potential adopters** are the students who need to submit their responses in both ht systems and also check for evaluations made by evaluator.

During the process of implementation, the user or sponsor was asked for feedback and reviews for the prototype versions developed. The user has approved and appreciated the results after seeing a demo of it. So, it is believed that it will be accepted immediately by the sponsor.
The major challenge likely to surface up would be from the students as they should try to understand the need for this system and issues faced by evaluator while following the manual process. A probable reason for resistance from students could be because they will have to submit the assignment separately at two different platforms. However, by communicating clearly the benefits of the system to them and enforcing them to use it, students are most likely to get on board with the sponsor and agents.

Appendix

Appendix A- User Manual

The user manual is divided into 3 parts, one for each type of user of this system. The three types of users are: Evaluators, Students and Admin.

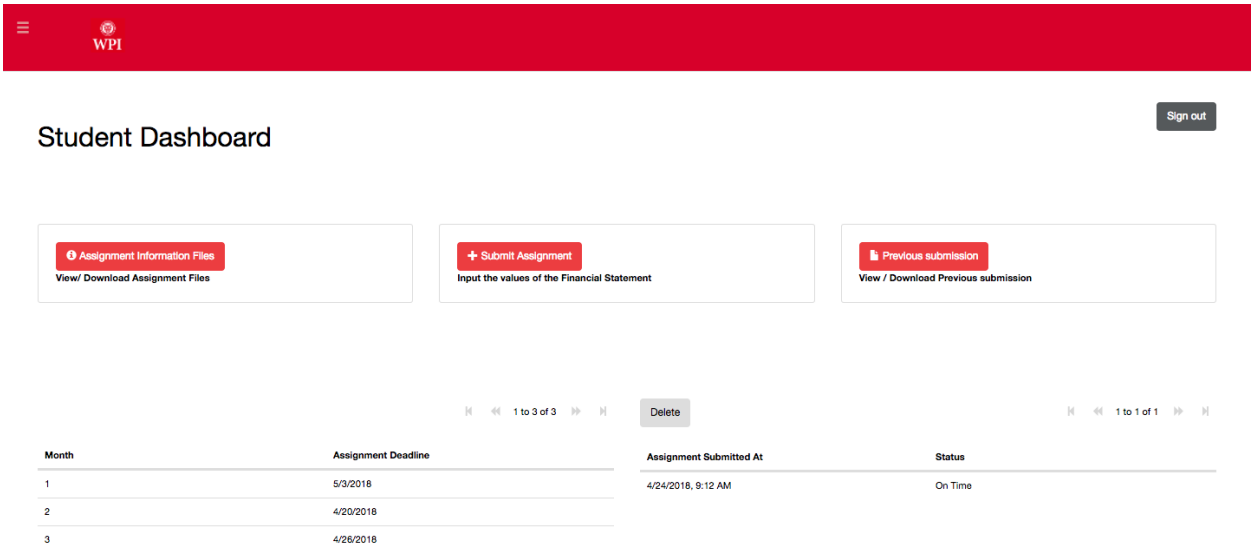


Figure 9: Student Dashboard

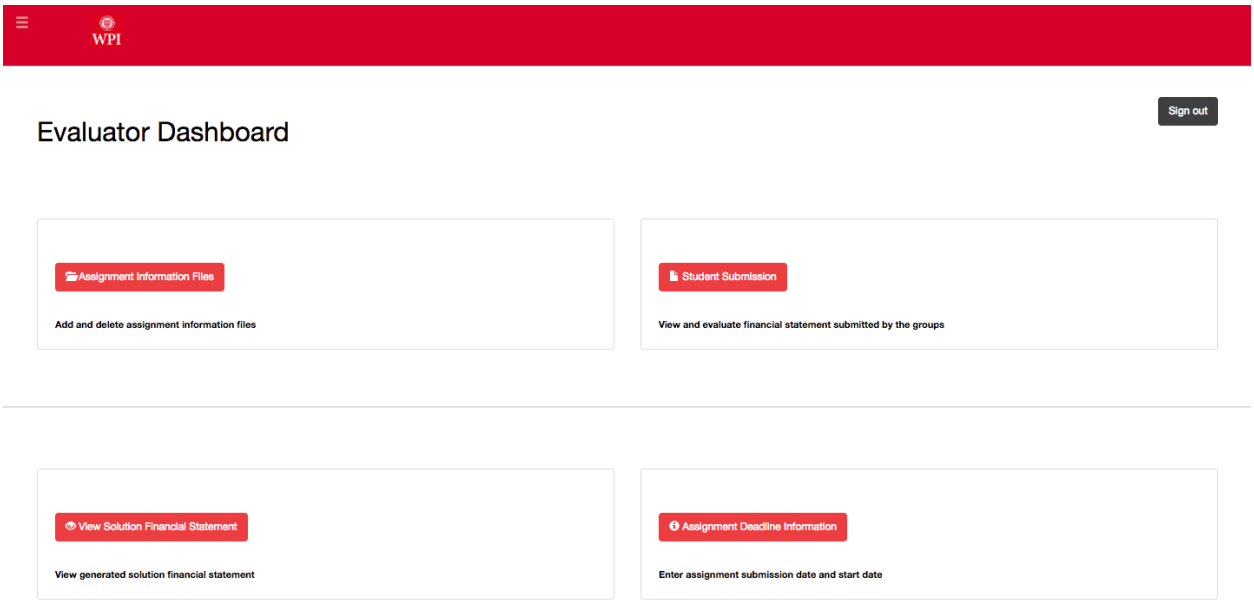


Figure 10: Evaluator Dashboard

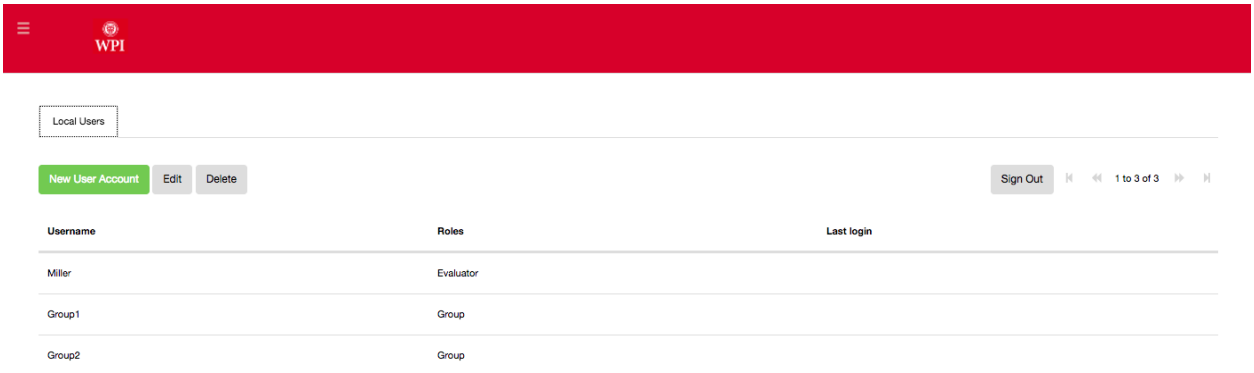


Figure 11: Admin Homepage

Evaluator

Manage Files

1. On the Dashboard Page, select "Assignment Information Files" icon.
    a. To search for a student, select the "Search" option and enter the file name and then select "Search" command on the right.
    b. To add a new file or a document, select "New" option, "Browse" the required file from the local disk, enter the file name and "Save" the file.
    c. To edit the file name or to add another file for the file name, select the required file from the list of files, select "Edit" option, modify the name or "Browse" the new file and "Save" the selection.
    d. To remove or delete the file from the system, select the required file from the list of files, select "Delete" option and give confirmation for deletion.
    e. To download a file from the system, select the required file from the list of files, select "Download" option, select "Save" option on the pop-up dialogue box.

View and Evaluate Assignments

1. On the Dashboard Page, select "Student Submission" icon.

2. To check the input values used by students to calculate their financial statement, select "Input Values".
    a. Select the month from the tabs for which the input values should be checked.
3. To check the financial statement prepared by the students, select "Financial Statements".
    a. Select the financial statement from the tabs for which comparison is to be viewed.
    b. To view the comparison between actual and expected click on "View Comparisons"
    c. To download the comparisons, select "Download Comparison Document" and select "Save" option on the pop-up dialogue box.

View Expected Financial Statement Solutions

1. On the Dashboard Page, select "View Solution Financial Statement" icon.
2. Select the Financial Statement from the tabs for which expected solution has to be viewed.
3. To download it in PDF format, select "Download Financial Statement" and select "Save" option on the pop-up dialogue box.

Set deadline for assignments

1. On the Dashboard Page, select "Assignment Deadline Information" icon.
    a. To search for a month's deadline information, select the "Search" option and enter the month number and then select "Search" command on the right.
    b. To add date and time details for each submission, select "New" option, enter the Month number, assignment start and end date and time, then "Save" the details.
    c. To edit the details of existing assignment information, select the required month from the list, modify the details and then "Save" the details.
    d. To delete details from the system, select the required record from the list and select "Delete" option, give confirmation.

Student Groups

View Assignment Instructions and information files

1. On the Dashboard Page, select "Assignment Information Files" icon.
    a. To search for a particular file, select "Search" option, enter the file name and select "Search" option at the right side.
    b. To download a file, select the required file from the list of files, select "Download" option, select "Save" option on the pop-up dialogue box.

Submit Assignment

1. On the Dashboard Page, select "Submit Assignment" icon.
    a. To enter the input values of financial statement/financial statement values, select "New" option, enter the values along with month number and "Save" the values.
    b. To edit the saved or submitted input values/financial statement values, select the month from the list for which values have to be modified, re-enter the values, "Save" the values.
    c. To delete the saved or submitted input values/financial statement values, select the month from the list and select "Delete" option, give confirmation.
    d. To view the previously submitted values, select "View Inputs" or "View Financial Statement" option, then select the required month from the tabs and view it.
    e. To confirm the submission, select the "Submit Assignment" option.

View previous submission and Download

1. Select the Financial Statement from the tabs for which has to be viewed.
2. To download the financial statement in the standard template, select "Download Financial Statement", select "Save" option on the pop-up dialogue box.

Check the status of previous submissions

1. On the Dashboard Page, the assignment status is available for viewing based on time submitted (on-time or late submission) for each month.

Admin

Create a New user account and credentials

1. To create a new user type and credentials, select "New User Account', enter the user name, select the user role from the list, create a password and re-enter the password, "Save" the values.
2. To search for a user name, select "Search" option, enter the user name and select "Search" option at the right side.
3. To edit an already added user account, select the required user name record, then select "Edit" option, make required changes and "Save" the details.
4. To delete a user account, select the required user name, then select the "Delete" option, give confirmation.

Appendix B- Deliverable 1

Project deliverable 1 report has been included in this section.

Automated System for U-Build Simulation Project

Project Deliverable #1

03/13/2018

Guided By: Prof. Bengisu Tulu

Prepared By:

Anuj Vyas
Manisha Chouhan
Nevetha Ramesh
Suraj Patil
Vishakha Jadhav

Contents

Abstract

This project has been initiated as part of the course MIS 573- Systems Design and Development. Through this project, we are developing a system for improving and automating existing process for assignment evaluation for U-Build simulation Project at WPI. We have used several techniques required for planning, analysis, design and implementation phases of Software Development Lifecycle for this project. Also, we have studied the necessary infrastructure, resources, and other essential aspects. This report covers all the required aspects for development of 'Automated System for U-Build simulation Project'.

1. Summary

1.1 Current System Process

The current system in place for submission and grading of the assignment can be studied through the following diagram:

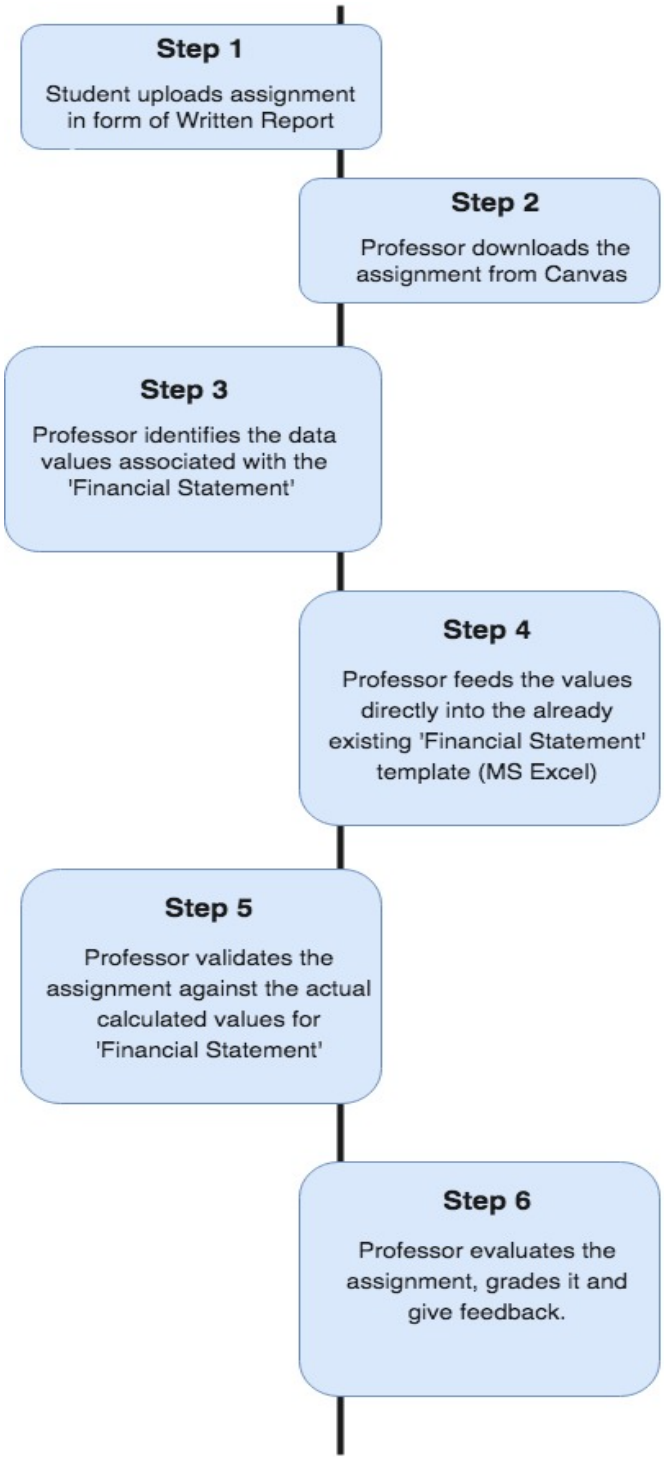


Illustration 1: Flow for current system

Financial statement is a collection of records of financial activities, position, results and cash flow of a business, person or other entity.

1.2 Proposed System Process

The proposed system is a web application where the students will provide the input values on a pre-designed form and system will generate a financial statement from these values by performing specific calculations.

The primary process is as follows:

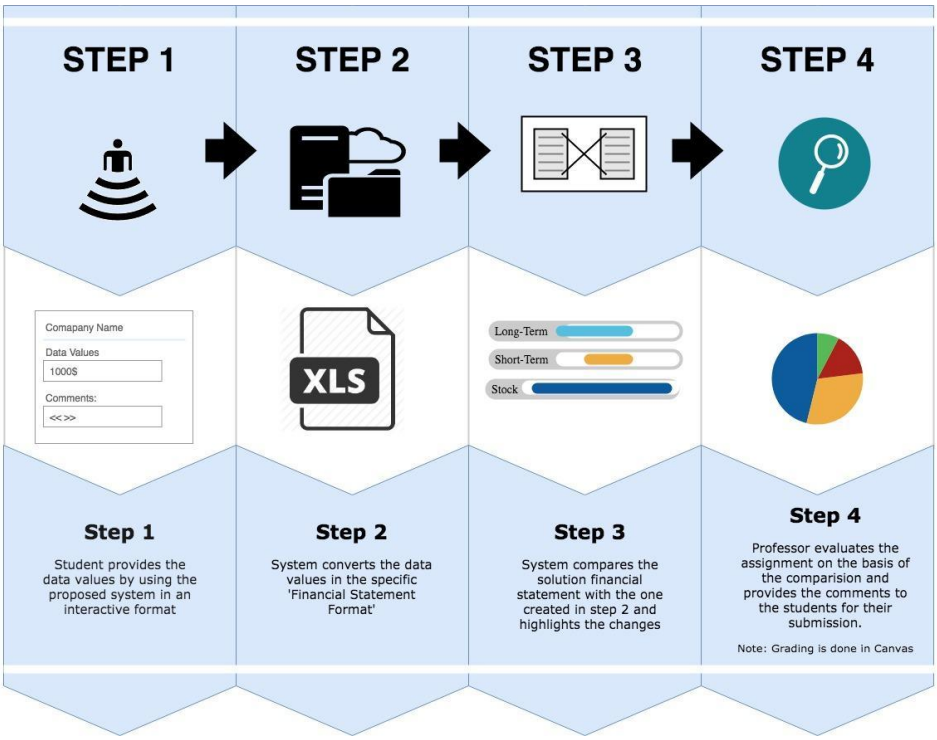


Illustration 2: Flow for Proposed System

2. System Request

System Request- Automated System for U-Build Simulation Project
Project Sponsor: Fabienne Miller, Associate Professor, Foisie Business School, WPI.
Business Need: This project has been initiated to automate the manual process of picking up the data from reports submitted as assignments by students for the U-build Project. The designed system will lead to increase in assignment evaluation efficiency, reduction in efforts and time. Also, it will decrease the human errors incurred while evaluating the assignments using the current system.
Business Requirements: Using a Web Application with a database, the system users will be able to store and retrieve the relevant data through an intuitive interface. The specific functionalities the proposed IT solution should have are as follows: ● Store the results of student solutions on pre-designed answer sheets. ● Transform answer sheets into standard financial statements. ● Establish a mechanism for comparing students' answer sheet with a standard solution sheet for financial statements. ● Highlight the answer deviations from expected results. ● Generate excel file for the financial statements submitted by students. ● Generate alerts for submission deadlines to students and notify late submissions to users. ● Accessibility to the previous submissions.
Business Value: We expect that, by automating the system for U-Build simulation project evaluation, there will be an increase in the overall work efficiency which will in-turn reduce the time, cost and efforts required. Professor will be able to make accurate and faster decisions on grading the assignments. Conservative estimates of tangible value to U-Build simulation project include the following: ● $ 6,600 form reduced Teaching Assistant salary expenditure per year ● 600 Hours saved from the reduction in evaluation time per year ● Return on Investment(ROI): 27.92% ● Break-even point: 2.7 years
Special Issues or Constraints: ● The system should be implemented before the start of the next semester. ● The system should align with WPI's academic policies and regulations.

3. Feasibility Analysis

3.1 Technical Feasibility

Project size: Medium; Risk - Moderate

- The system will be developed by a group of 5 members who are new to the technology platform with not much hands-on experience.
- The complete end-to-end system will be attempted to be delivered as per the project deadline.

Familiarity with technology: Low; Risk - High

- The system will be developed for users who might not have complete expertise in technology for which, a technical or maintenance team will be available to guide them for using the application.
- The software used to develop the application is newly introduced to the team who have a beginner level of expertise for the same.
- The support or maintenance staff will be required to learn and understand the technology before they can start working on top of the existing application.

Familiarity with Application: Medium; Risk - Moderate

- The project team has been introduced to the working of the existing application and have access to the documents about the current system but, are yet to have a complete understanding of it.
- The new system, from the users' point of view, will be similar in terms of the process being followed. But, the platform is going to be different.
- The new system will provide detailed documentation as well as a tutorial to use and better understand how the application works.

Compatibility: High; Risk - Low

- As per the requirements, the to-be system will be developed as a desktop application and can easily be integrated with the existing technology, already in use.
- The to-be application will be compatible with internet browsers - Internet Explorer Edge, Chrome, and Firefox, etc.

3.2 Organizational Feasibility

- Strategic Alignment:

With the help of the new system, the professor will not have to do the evaluation manually. Instead, the new system will satisfy the user's goal to accelerate the process of grading the students' assignments.

- Stakeholder analysis:
  The developed application can be accessed by faculty members within WPI who use the same simulation project as assignments in their courses. The to-be system will be used by the students to submit their financial statements and will be used by the Professors and Teaching Assistants, i.e., evaluators, for evaluating them.

3.3 Economic Feasibility

	2018	2019	2020	2021	Total
Benefits					
Increased Time Savings		$ 12,000	$ 13,200	$ 14,520	
Reduced TA Salary		$ 5,000	$ 5,000	$ 5,000	
Increased Reputation		$ 25,000	$ 27,500	$ 30,250	
Total Benefits		**$ 42,000**	**$ 45,700**	**$ 49,770**	**$ 1,37,470**
Present Value of total benefits		**$ 40,384.62**	**$ 43,942.31**	**$ 47,855.77**	**$ 1,32,182.69**
Development Cost					
Personnel costs	$ 30,600	$ -	$ -	$ -	
Hardware	$ -	$ -	$ -	$ -	
Software	$ -	$ -	$ -	$ -	
Total Development costs	**$ 30,600**	**$ -**	**$ -**	**$ -**	**$ 30,600**
Operating costs					
Personnel Costs		$ -	$ -	$ -	
Hardware		$ -	$ -	$ -	
Software Licensing		$ 22,500	$ 22,500	$ 22,500	
Maintenance Costs		$ 3,000	$ 3,120	$ 3,244.80	
Total operational cost		**$ 25,500**	**$ 25,620**	**$ 25,744.80**	
Total costs	**$ 30,600**	**$ 25,500**	**$ 25,620**	**$ 25,744.80**	**$ 1,07,464.80**
Total benefits-Total costs	**$ (30,600)**	**$ 16,500**	**$ 20,080**	**$ 24,025.20**	**$ 30,005.20**
Cumulative net cash flow	**$ (30,600)**	**$ (14,100)**	**$ 5,980**	**$ 30,005.20**	

ROI Rate of investment	**27.92%**	
Break Even Point	**2.7 years (2 years of negative cumulative cash flow + [20080-5980]/20080=0.70)**	

Table 1: Cost Benefit Analysis for Automated system for U-Build Simulation Project

Assumptions:

- Maintenance cost in operational costs will increase at the rate of 4% per year.
- There will be 10 % increase in the increased time savings and increased reputation on a yearly basis.
- Intangible benefits have been assigned a value by evaluating the actual monetary value being saved.
- For instance:
  - By automating the current system, the professor will save 5 hours/ week of time which she can utilize for other projects and research.
  - Due to increased reputation, more research projects are being sponsored leading to more funds to WPI.
- All development and operational costs are assumed as per the following criteria after gathering data from trusted sources:
  - Consultation Fee for Systems Analyst/ Designer is considered as 40 $/Hour, with a scope of working for a minimum of 80 Hours.
  - Development Fee for Developer is considered as 45 $/Hour, considering three developers working for 18 days for 8 Hours/ Day
  - Users are assumed not to have proper technical knowledge. Hence, requires training for operating the current system.
  - Software licensing cost will be charged after the first year which is $ 1875/month
  - Hardware and software values are $ 0 because we have access to the software (Mendix) for one year and WPI fulfils all the hardware requirements. However, licensing cost for subsequent years is considered in operational costs.

3.4 Implementation of P.I.E.C.E.S Framework:

P - The need to improve performance

The response time of the to-be application will be minimal as compared to the as-is system. Students, professors, and admins will be able to retrieve the pages with better response time. The to-be system will also provide better throughput compared to the previous methodologies used to evaluate the assignments.

I - The need to improve data and information

The as-is system has data distributed over various sources which make the assignment evaluation a tedious process. The to-be system will allow the users to capture all required data at one destination in a readily usable format.

E - The need to improve economics, control costs, or increase profits

The to-be system will help control costs since the system will be automated, the number of Teaching Assistants required would also be minimal compared to the as-is system where professors and students had to evaluate/grade the assignments manually taking more efforts and time.

C - The need to improve control or security

The system monitors the user's login credentials, and only the admins, evaluator will be able to access the financial statement solutions sheet and assignment submissions so that students do not get access to the assignment solutions.
The student groups will have restricted access, i.e., through group login, students will not be able to access the submissions/solutions of other groups.

E - The need to improve efficiency of people and processes

The system will integrate the assignment submitted by the students and compares them with the solution sheet. The system ensures that all the assignments will be evaluated according to the predefined formulas so that there are fewer chances of error and improve the precision of the evaluation system.

S - The need to improve service to customers and suppliers

The to-be system will provide consistent and reliable results to the evaluators. Also, the system is flexible, i.e., we can add additional features if required with minimal costs.

4. Project Charter

This Project Charter reflects how we will work and coordinate as a team, our meeting times, mode of communication and how we are going to share the information and essential and consequences for not following the guidelines. This charter is required to be followed by every team member for the successful completion of this project.

Project Meetings:

Internal Team Meeting:

- The team should meet twice in a week to discuss, brainstorm and share the status of their assigned responsibilities. If the team member is not available, prior information must be provided to team lead. In this case, a virtual meeting can be scheduled to accommodate every member in the team.
- Every member should communicate wisely and make other members understand by providing facts.
- The responsibility to take minutes of the meeting (MOM) will be rolling within the team in a cyclic order. Every member should understand the process involved in maintaining the MOM.

Meeting with Project Sponsor:

The weekly meeting is organized as per the availability of the project sponsor. Formal meeting invites are sent through email via outlook.

Mode of Communication:

The team communicates with each other through Hangouts Group and Group Email ID - u-build@WPI.EDU over Microsoft Outlook.

Information Sharing:

Shared google drive is created and is accessed by the team members only. The information received from the project sponsor is properly documented for future reference. The confidential information shared by the professor will not be shared on any unsecured, informal platform since it will be considered to be a violation of non-disclosure agreement.

Team Coordination:

- Every member of the team is responsible for preparing for the meeting with some inputs that will take the project to the subsequent level.
- Developing an agenda with proper questions that will include the discussion questions that enable team members to discuss the issues and To-Do list specific to meetings.
- Decisions will be made based on majority and consensus of the team members on the basis of evident facts.
- Discussion on the updates done by each team member to check if the project is moving on the right path.
- The tasks and time required for completion of tasks are identified through project plan, and team members are expected to follow the plan properly.
- Team conflicts are addressed by discussing the issue within the team in team meetings.
- Team members are not permitted to use others work or plagiarize work and violate rules of WPI academic integrity policy.

If a member fails to meet this minimum level, he or she will be subject to the following consequences:

- The first time a team member fails to meet his or her responsibilities, the team will meet and give a verbal warning. This warning will include three components: the section of the Team Performance Charter that the person violated, suggestions for improving his or her performance, and metrics to evaluate improved performance in the future.
- The second violation by a team member should result in a written warning with a copy emailed to Prof. Bengisu Tulu. This should summarize the account of the problem and remedial actions that have been attempted.

5. Team Roles

Role	Skills	Responsibilities	Assigned to
Project Manager	• Knowledge of Software Development Life Cycle • Knowledge of PERT and GANTT Chart • Hands on experience of project management tools	• Creating work plan and staffing the project. • Organising and motivating a project team • Controlling team and time management • Cost estimating and developing the budget	Manisha Chouhan
Systems Analyst	• Efficient planning and execution • Database design expertise • Strong programming/coding	• Broad knowledge of hardware, software and programming • Planning and negotiating, awareness of the business	Suraj Patil

Programmer	• Experience in C++, JAVA, HTML, CSS, XML • Documenting requirements • Debugging, testing, building and deployment of code.	• Problem solving abilities • Understanding of algorithms and data structures • Business sense	Anuj Vyas, Vishakha Jadhav
Quality Analyst	• Automation testing • SQL, PL/SQL • SAP Enterprise Suite • Documentation	• Analytical thinking • Ability to code automated tests	Nevetha Ramesh

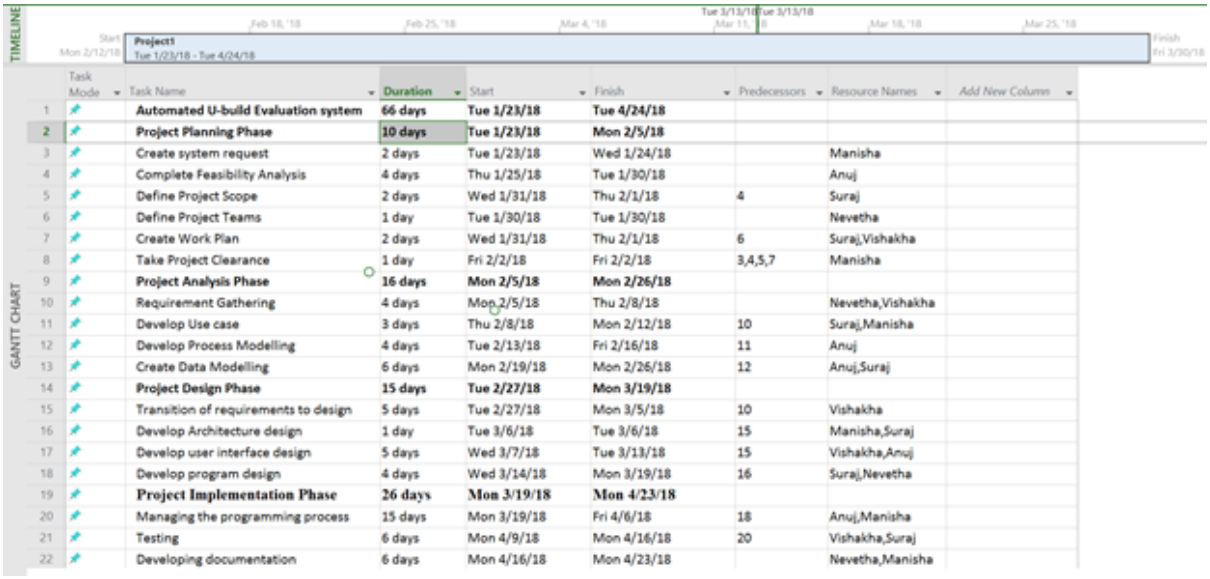
Table 1: Roles assigned to the team members as per the skills.

Role	**Constraints**
Project Manager	• Miscommunication or unclear instructions to team members can change client's original requirements • Managing project within budgeted cost
Systems Analyst	• Interpreting client requirements and instructing them to the team
Programmer	• Design or Development of the project within a specified time • Keeping up to the expectations of the management team
Quality Analyst	• Creation and execution of test cases as per the deadline • Validating results, testing application from user's perspective.

Table 2: Constraints associated with the roles.

6. Project Plan

6.1 Work Plan:



TIMELINE

Start Mon 2/12/18 — Project1 Tue 1/23/18 - Tue 4/24/18 — Finish Fri 3/30/18

GANTT CHART

	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names	Add New Column
1		**Automated U-build Evaluation system**	**66 days**	**Tue 1/23/18**	**Tue 4/24/18**			
2		**Project Planning Phase**	**10 days**	**Tue 1/23/18**	**Mon 2/5/18**			
3		Create system request	2 days	Tue 1/23/18	Wed 1/24/18		Manisha	
4		Complete Feasibility Analysis	4 days	Thu 1/25/18	Tue 1/30/18		Anuj	
5		Define Project Scope	2 days	Wed 1/31/18	Thu 2/1/18	4	Suraj	
6		Define Project Teams	1 day	Tue 1/30/18	Tue 1/30/18		Nevetha	
7		Create Work Plan	2 days	Wed 1/31/18	Thu 2/1/18	6	Suraj,Vishakha	
8		Take Project Clearance	1 day	Fri 2/2/18	Fri 2/2/18	3,4,5,7	Manisha	
9		**Project Analysis Phase**	**16 days**	**Mon 2/5/18**	**Mon 2/26/18**			
10		Requirement Gathering	4 days	Mon 2/5/18	Thu 2/8/18		Nevetha,Vishakha	
11		Develop Use case	3 days	Thu 2/8/18	Mon 2/12/18	10	Suraj,Manisha	
12		Develop Process Modelling	4 days	Tue 2/13/18	Fri 2/16/18	11	Anuj	
13		Create Data Modelling	6 days	Mon 2/19/18	Mon 2/26/18	12	Anuj,Suraj	
14		**Project Design Phase**	**15 days**	**Tue 2/27/18**	**Mon 3/19/18**			
15		Transition of requirements to design	5 days	Tue 2/27/18	Mon 3/5/18	10	Vishakha	
16		Develop Architecture design	1 day	Tue 3/6/18	Tue 3/6/18	15	Manisha,Suraj	
17		Develop user interface design	5 days	Wed 3/7/18	Tue 3/13/18	15	Vishakha,Anuj	
18		Develop program design	4 days	Wed 3/14/18	Mon 3/19/18	16	Suraj,Nevetha	
19		**Project Implementation Phase**	**26 days**	**Mon 3/19/18**	**Mon 4/23/18**			
20		Managing the programming process	15 days	Mon 3/19/18	Fri 4/6/18	18	Anuj,Manisha	
21		Testing	6 days	Mon 4/9/18	Mon 4/16/18	20	Vishakha,Suraj	
22		Developing documentation	6 days	Mon 4/16/18	Mon 4/23/18		Nevetha,Manisha	

Illustration 3: Work plan having tasks, duration, start date, end date and task allocation.

6.2 Gantt Chart

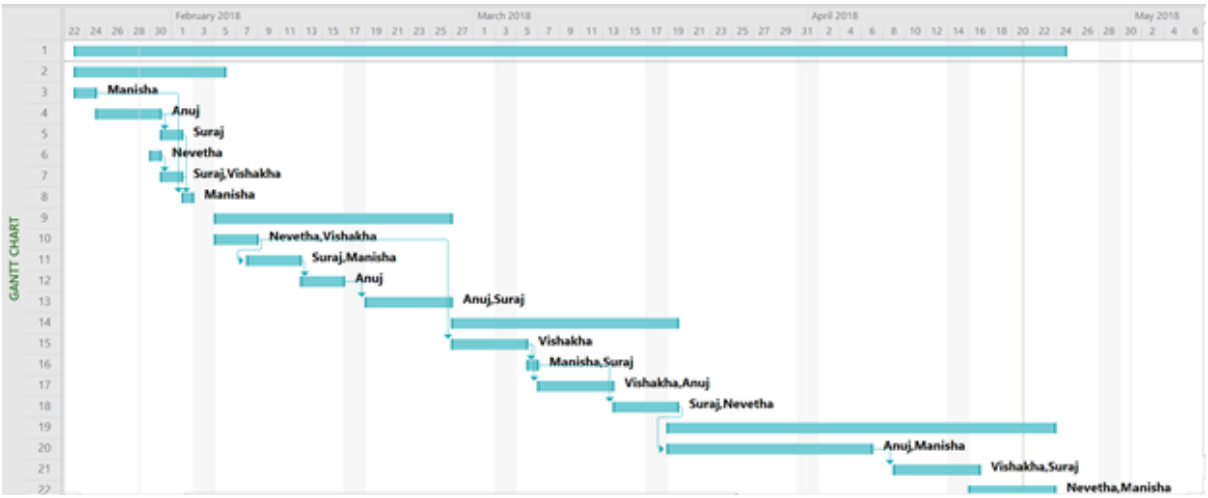


Illustration 4: Gantt Chart for the created work plan

Note: Please zoom in images for illustration 3 and 4 for clarity.

7. Analysis Strategy

7.1 Requirements Elicitation Techniques Used

7.1.1 Interviews

We conducted one to one meeting with the Project Sponsor, Professor Fabienne Miller.

Steps:

- Setting up a meeting as per the availability of the project sponsor.
- Providing a list of questions with the agenda for the meeting to ensure the proper insights from the project sponsor.
- Interviewed the Project Sponsor with the prepared questions.
- Documenting the answers or insights for future reference.

This technique was beneficial because of the following reasons:

- A better understanding of the as-is system and current process in place.
- Gathered more information about the 'Finance' domain.
- Gathered the actual requirements for the to-be system as a result of several analysis strategies used.

The questions being asked to the Project Sponsor in interviews and through emails are as follows:

- Can you explain the current process being followed for the assignment submission?
- Can you explain the current process being used for the evaluation of assignments?
- What can be understood from the 'Financial Statement' for the U-Build simulation process?
- Is there is any specific format for the 'Financial Statement'? If yes, can you provide the template?
- Can you explain the 'Financial Statement' for U-build simulation project and the formulas used for the calculation of the values?
- Do you have any specific requirement for the 'Look and feel' of the application?
- Do you want an alert system in place for the sending an alert to the students for assignment submission before a specific time duration?
- Do you need any provision to provide 'Feedback' or 'Comments' to the submissions?
- Do you want to download or maintain a local copy of the student submissions and the comparison between the submitted and actual solution?
- How many types of users do you think the proposed IT system will have?
- Do you need to provide access restriction for the various users for the system for their login? If yes, what are the restrictions?
- Do you want the developed web application to be readily workable in all the browsers? If no, do you have any specific requirement for browser?
- Do you want a provision to change the assignment details, i.e., the assignment files, standard solution for the financial statements, assignment instructions? If yes, is it going to be a file upload and deletion?
- Who is going to be benefitted from this application development? Whole department/ Business School/ WPI?
- If the developed application is shared with other universities who use the same project, how much should be the charge of it?
- In an attempt to convert the in-tangible values to tangible one, can provide monetary value associated with the reduction in your effort for evaluation of assignments? (Optional/ Assumed Value will also work)

7.1.2 Document Analysis

The documents have been provided after signing a formal Non-Disclosure Agreement with the Project Sponsor.

List of Documents Received:

- Assignment Instructions and Memorandums.
- The solution sheet for Financial Statement.
- Sample Student Submission.

Note:

We cannot disclose the further information as we are under the formal Non-Disclosure Agreement.

This technique was beneficial for us because of following reasons:

- Assignment Instructions and Memorandums shows the clear picture of what is expected from the assignments submitted by the students.
- Solution sheet for Financial Statement provides the actual data values and formulas required for the various calculations.
- The document analysis has resulted in the assumptions for the cost-benefit analysis for this project. We have identified the intangible benefits of the to-be system using the provided documentation.
- These documents provided the actual information required to study the as-is system as well.

7.2 Requirement Analysis Strategies Used

Understanding the as-is system was essential for us as it is going to form the foundation for the to-be system. Also, the proposed project is complicated concerning domain knowledge, i.e., it requires the proper understanding of developing 'Financial Statements' based on the provided information in the form of memos.

7.2.1 Problem Analysis

After having a good knowledge of the as-is system, we have identified the problems associated with it by conducting Interviews with the Project Sponsor.

We came up with the following Problems associated with the As-Is system:

- The manual process of evaluation is tedious and time taking, i.e., to identify the associated data values evaluator has to read the whole report submitted by the students.
- The current system requires more efforts regarding the conversion of identified data values into financial statements.
- Sometimes, students provide the incorrect data values which are difficult to locate when written in report format during the initial evaluation.
- Maintaining the records for the previous submissions associated with Financial Statements is difficult as students submit only the written reports as part of their assignment submission.

Solutions identified for the above problems addressed by the to-be system are as follows:

- The manual process is converted into an automated system where the students insert the data values in the pre-specified format.
- The system generates the Financial Statement automatically as per the data values entered by the students.
- The system can generate the difference between the standard FS solution and generated FS solution automatically.

- The system stores the generated FS solution and the comparison sheet for future reference.

7.2.2 Duration Analysis

The duration analysis considers the comparison of the amount of time taken by the as-is system and to-be system for the process. We have considered the primary process of 'Assignment Evaluation' for this.

As-Is System:
Approximate time taken for evaluation = 2.5 Hours/ Evaluation.
Total number of evaluations per week = 10
Total number of hours spent weekly evaluation = 25 Hours

To-Be System:
Approximate time taken for evaluation = 0.5 Hours/ Evaluation.
Total number of evaluations per week = 10
Total number of hours spent weekly evaluation = 5 Hours

Hence, we concluded that minimum of 20 Hours of time taken for the evaluation in the as-is system can be saved by implementing to-be system.

7.2.3 Outcome Analysis

The to-be system provides the following intangible values:

- Faculty time saved.
- Potential to share the project with other faculty members at other universities (current project is too labour intensive) thus enhancing the reputation of WPI.

Hence, we can evidently say that the to-be system is Business Process Improvement and Automation as current business process is understood, the opportunity for enhancement is identified and the process will be suitably redesigned to improve the workflow and automate few processes in the current system.

8. Requirements

All the functional and non-functional requirements are divided into four sprints which are as follows:

Sprint 1:

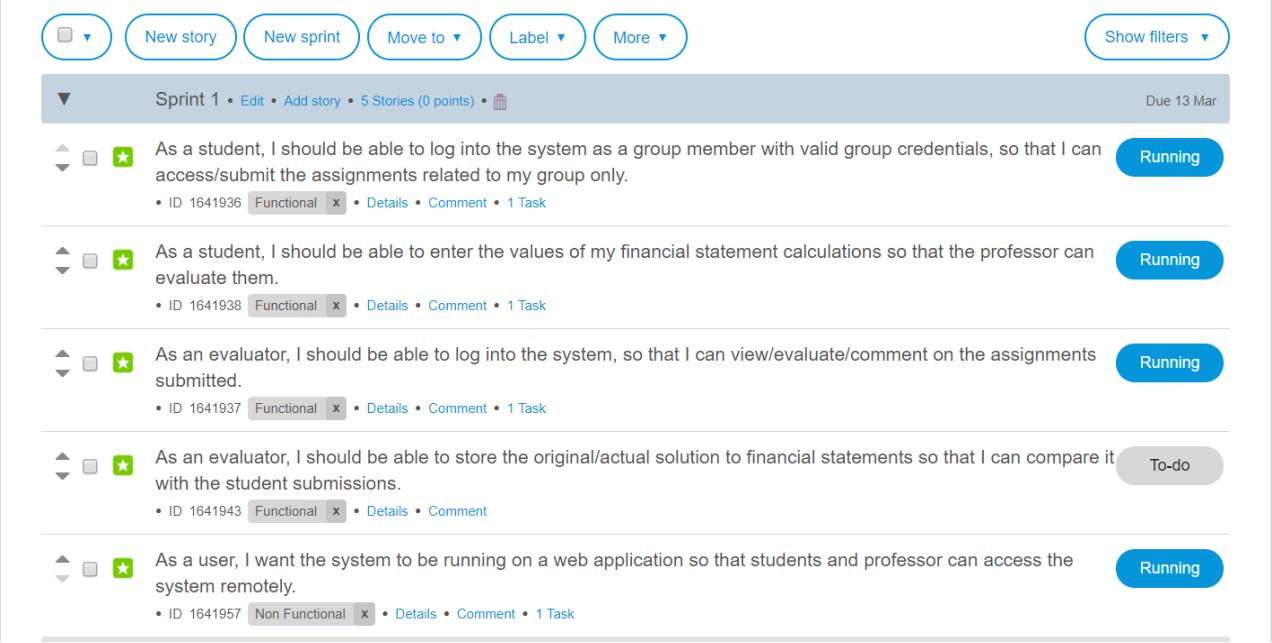


Illustration 4: Five user requirements/ stories to be implemented as part of Sprint 1

Sprint 2:

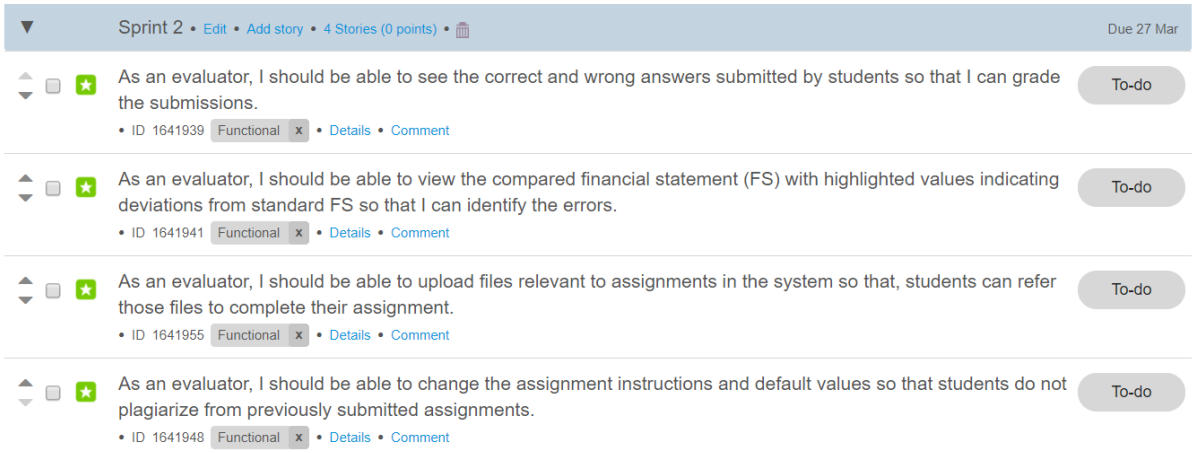


Illustration 5: Four user requirements/ stories to be implemented as part of Sprint 2

Sprint 3:

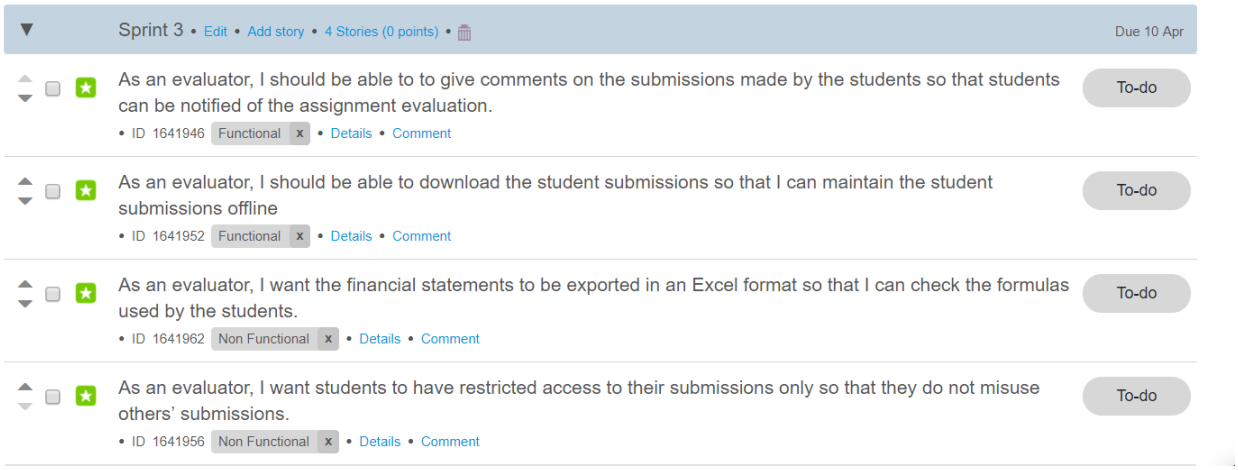


Illustration 6: Four user requirements/ stories to be implemented as part of Sprint 3

Sprint 4:

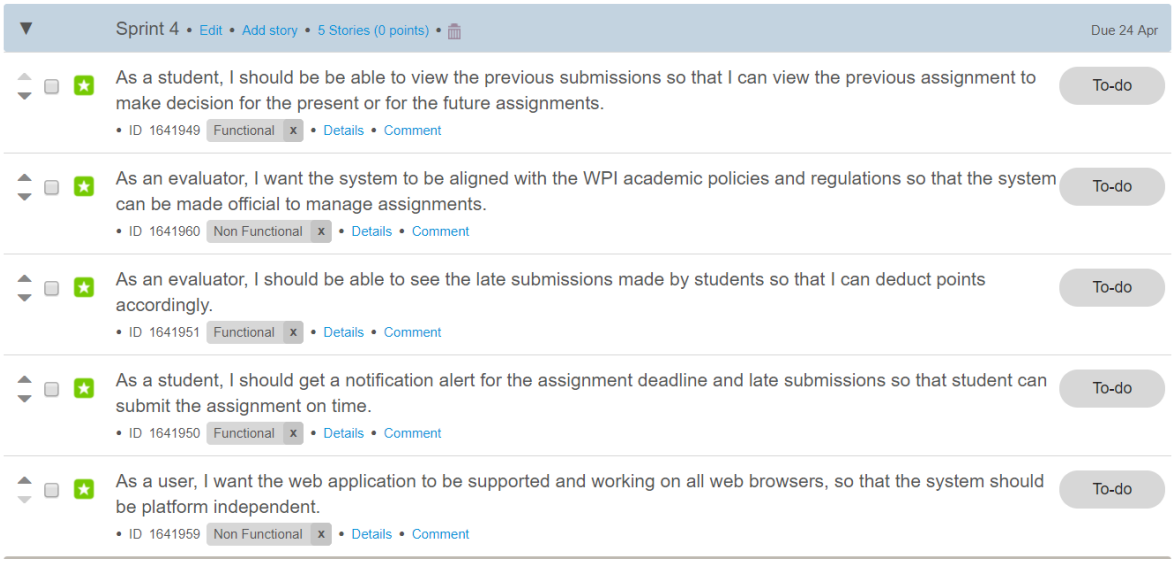


Illustration 7: Five user requirements/ stories to be implemented as part of Sprint 4

9. Use Cases

9.1 For AS-IS System

9.1.1 USE CASE I

Use Case Name: Submitting an assignment	**ID:** UC-1	**Priority:** High
Actor: Student		
Description: Student uploads the Financial statement report on Canvas system for his/her group		
Trigger: Student requests for assignment submission **Type:** External		
Preconditions: 1. The system is up and running. 2. Student has logged in to the system with valid credentials and has navigated to assignment submission page. 3. The assignment instruction file should be uploaded on canvas for students.		
Normal Course: 1.0 Evaluating assignment 1. Student selects submit assignment option. 2. System displays option to select the files to be uploaded from student's local system. 3. Student uploads the assignment file. 4. System requests a submission confirmation from student. 5. Student confirms the submission. 6. System checks the assignment deadline and displays a message based on on-time or late submission. 7. System displays 'turned-in' message for on-time submission. 8. System displays an option to re-submit the assignment.		**Information for Steps** Submit request List of local files Files to be uploaded Confirmation Request Confirmation Turned-in message Re-submit option
Alternate Course: (Branch at Step 6) 1. System displays 'late submission' message for post deadline submissions.		Late submission message
Postconditions: 1. Database is updated with the latest assignment file with submission related details. 2. Evaluator is notified on canvas for the submission made.		

Summary

Inputs	**Source**	**Outputs**	**Destination**
Submit request	Student	List of local files	Student
Files to be uploaded	Student	Confirmation request	Student
Confirmation	Student	Turned-in message	Student
		Resubmit option	Student
		Late submission message	Student

9.1.2 USE CASE II

Use Case Name: Evaluating the assignment	**ID:** UC-2	**Priority:** High
Actor: Evaluator		
Description: The Financial Statement (FS) submitted by students will be downloaded by the evaluator. The evaluator checks the submitted values with the standard solution to provide feedback and grade the submission.		
Trigger: Evaluator requests for students' assignment for evaluation **Type:** External		
Preconditions: 1. The system is up and running. 2. Evaluator is logged in the system with valid credentials. 3. Database is updated with the student submissions.		
Normal Course: 1.0 Evaluating the assignment 1. System displays the list of assignment submission. 2. Evaluator selects the assignment to be evaluated. 3. Evaluator selects to download the assignment from the system. 4. Evaluator copies the values from the report submitted. 5. Evaluator pastes it on an excel sheet which has predefined formulas and a template. 6. Evaluator converts the excel sheet into a financial statement. 7. Evaluator compares the financial statement with a standard financial statement solution. 8. Evaluator provides feedback and grading for the submitted assignment on the canvas system.	**Information for Steps** List of files Selection Download Selection Calculated values Template Financial Statement Comment and grade	
Alternate Course: (Branch at Step 6) 1. System displays 'late submission' message for post deadline submissions		
Postconditions: 1. Database is updated with grades and feedback to view on canvas.		

Summary

Inputs	**Source**	**Outputs**	**Destination**
Selection	Evaluator	List of files	Evaluator
Download selection	Evaluator	Financial Statement	Evaluator
Calculated values	Student		
Template	Evaluator		
Comment and grade	Evaluator		

9.2 For TO_BE System

9.2.1 USE CASE I

<table>
<tr><td colspan="2">Use Case Name: Sending assignment deadline alert notifications</td><td>ID: UC-1</td><td>Priority: High</td></tr>
<tr><td colspan="4">Actor: System</td></tr>
<tr><td colspan="4">Description: The deadline specified by the professor for the assignment will be taken by the system as a trigger point to send the alert to the students, 12 hours before the actual deadline for those who have not submitted the assignment yet.</td></tr>
<tr><td colspan="4">Trigger: The timer which is 12 hours prior to the submission deadline is hit
Type: Temporal</td></tr>
<tr><td colspan="4">Preconditions:
1. The assignment deadline details are fed to the system.
2. The assignment relevant files data store available in the database
3. The system is online and up and running
4. All the student email ID’s are available in the database</td></tr>
<tr><td colspan="3">Normal Course:
1. Sending assignment deadline alerts to the students
1. The system will check for the assignment submissions made by groups till the trigger time
2. The system will fetch the email ID’s of the students in the groups who have not submitted the assignment
3. System will send an alert saying, ‘You have the U-Build assignment pending at (deadline)’</td><td>Information for Steps

Assignment submissions

Student email ID’s
Alert Notification</td></tr>
<tr><td colspan="4">Postconditions:
1. The system has no trigger set for the assignment</td></tr>
<tr><td colspan="4">Summary</td></tr>
<tr><td>Inputs</td><td>Source</td><td>Outputs</td><td>Destination</td></tr>
<tr><td>Student email ID’s

Assignment submissions</td><td>Student Information datastore

Financial Statement records datastore</td><td>Alert Notification</td><td>Student</td></tr>
</table>

9.2.2 USE CASE II

Use Case Name: Submit an assignment	**ID**: UC-2	**Priority**: High
Actor: Student (Group Login)		
Description: Student groups submit the final calculated and assumed values of financial statement in a pre-designed format.		
Trigger: Student wants to submit the assignment.		
Type: External		
Preconditions: 1. The system should be up and running. 2. The student should be logged into the system with valid group credentials. 3. The student is on the assignment submission page. 4. Assignment deadline and files are available in the system.		
Normal Course: 1.0 Compare values with standard financial statement values. 1. System displays the assignment form 2. Students enters the assumed and calculated values on the form 3. Student selects to submit assignment 4. System generates and displays the financial statement 5. Student confirms the assignment submission 6. System checks the assignment deadline time from the datastore to display confirmation message based on-time or late submission (Alternative course) 7. System displays 'turned-in' message for on-time submission		**Information for steps:** Assignment form Values Selection Financial statement Confirmation Turned-in message
Alternative Course: (branch at step 5) 1.1 Student has submitted the assignment after the deadline. 1. System displays 'late submission' message for post deadline submissions		 Late submission message
Post conditions: 1. Datastore is updated with the generated financial statement and submission related details.		

Summary **Inputs**	**Source**	**Output**	**Destination**
Values	Student	Assignment form	Student
Selection	Student	Financial statement	Student
Confirmation	Student	Turned-in message	Student
Deadline time	Assignment detail datastore	Late-submission message	Student

9.2.3 USE CASE III

Use Case Name: Comparing assignments using standard Financial statement	**ID**: UC-3	**Priority**: High
Actor: System		
Description: System compares the generated financial statement with a standard financial statement in the database and updates the datastores.		
Trigger: System has generated a Financial statement for the submitted input values		
Type: Temporal		
Preconditions: 1. The system should be up and running. 2. The values entered by the students should be stored which the system can access. 3. The database is available and has previously stored files (if any). 4. Database is updated with the generated group-wise financial statements		
Normal Course: 1.0 Compare values in generated financial statement with standard financial statement. 1. System access the generated and standard financial statement from datastores. 2. The system checks whether there is any deviation or not from the actual values. (Alternative course 1.1) 3. If there are deviations, the system highlights the deviations. 4. The datastores are updated with the compared and highlighted financial statement.		**Information for steps:** Financial statement Highlighted values Compared Financial statement
Alternative course: 1.1 No deviations from the standard financial statement in the generated one. 1. The system updates the datastore with the financial statement without any highlight.		Compared Financial statement
Post conditions: 1. The compared and highlighted financial statements are stored in the database.		
Exceptions: E1. There is no standard financial statement in the system's database yet. 1. The system does not compare the values. 2. System stores the financial statement directly without any highlights with an indication that it is yet to compared.		

Summary **Inputs**	**Source**	**Output**	**Destination**
Financial statement	Financial statement datastore	Highlighted values Compared Financial statement	Evaluator Financial statement datastore

9.2.4 USE CASE IV

Use Case Name: Evaluating the assignment	**ID:** UC-4	**Priority:** High
Actor: Evaluator		
Description: The Financial Statement (FS) submitted by students with highlighted deviations from standard FS solution will be provided to the evaluator for evaluation. The evaluator can also view the standard FS solution at the same time to cross check. After evaluation evaluator may provide some comments or feedback to the students.		
Trigger: Evaluator requests for students' assignment for evaluation **Type**: External		
Preconditions: 1. Evaluator has logged in the system with valid credentials. 2. The correct standard FS solution is loaded in the system. 3. The system has the FS with the highlighted deviation values from the standard FS solution.		

Normal Course:	**Information for Steps**
1.0 Evaluating the assignment	
1. Evaluator selects the student submission evaluation option	Evaluation selection
2. The system displays the list of all the groups for selection	List of groups
3. Evaluator selects the group he/she wants to evaluate the assignment.	Group selection
4. The system will display the list of all the assignments, both evaluated and not evaluated; submitted by that group	List of assignments
5. Evaluator selects the assignment to be evaluated	
6. System displays the FS with highlighted deviation values from the standard FS solution	Assignment selection Student FS with highlights
7. System also provides a copy of the standard FS solution to the evaluator for referring	Standard FS solution
8. Evaluator after evaluation will enter feedback comments if he/she wants to	
9. The system will add the comments in the comments section which can be viewed by the students.	Comments Updated comments

Postconditions:

1. The system stores the comments given by the professor for each assignment submission which can be viewed by all the group members of that submission.

Exceptions:

E1: There is not a single assignment submitted by the group

1. The system will show a message saying 'No assignment submitted by group #'
2. The system will go back to the previous page for selection of group where the evaluator can select another group

Summary

Inputs	**Source**	**Outputs**	**Destination**
Evaluation selection	Evaluator	List of groups	Evaluator
Group selection	Evaluator	List of assignments	Evaluator
Assignment selection	Evaluator	Student FS with highlights	Evaluator
Comments	Evaluator		

		Standard FS solution Updated comments	Evaluator datastore

9.2.5 USE CASE V

Use Case Name: Access previously submitted assignments	**ID**: UC-5	**Priority**: Medium
Actor: Student /Evaluator		
Description: The actor views the history of group-specific submissions and selects the required submission which needs to be downloaded in Excel format.		
Trigger: User wants to access the submission history to view or download. **Type**: External		
Preconditions: 1. User has logged in as student/Evaluator with valid credentials. 2. The actor is on the submission history page. 3. The system has stored group-specific submissions made by students.		
Normal Course:		**Information for Steps:**
1.0 Access previously submitted assignments by student or evaluator		
1.System checks whether the type of user is an evaluator or student group based on login credentials. (Alternative Course 1.1)		User type
2. The login credentials match with a student group in the database.		
3. System displays list of assignments submitted only by that group		Assignment List
4. User selects the required the assignment number.		Assignment number
5. 6. System gives option whether to view or download the assignment (Alternative course 1.2)		View/download option
7. User selects the option to view the assignment.		View request
8. System displays the assignment.		Assignment display
Alternative Course:		
1.1 The login credentials matches with an evaluator in the database (Branch at step 1)		
1.System displays list of group numbers.		Group numbers list
2. Evaluator selects the group number from the list.		Group number
1.2 User selects the option to download the assignment (Branch at step 6)		Download request
1. System converts the financial statement to an Excel format.		Financial Statement
2. System provides offline copy of the assignment.		Excel File
Post conditions: 1. The Excel is downloaded on the user's storage device.		
Exceptions: E1: There is no previous submission made by the group. 1. System displays an Error message. "No submissions made". **2.** System displays an option to go to Home page.		

Summary **Inputs**	**Source**	**Output**	**Destination**
Login credentials	User datastore	Assignment List	User
Assignment Number	User	View/download option	User
View request	User	Assignment display	User
Group number	Evaluator	Group numbers list	Evaluator
Download request	User	Excel File	User
Financial Statement	Submission Datastore		

9.2.6 USE CASE VI

Use Case Name: Update relevant files	**ID**: UC-6	**Priority**: High
Actor: Evaluator		
Description: The evaluator updates the relevant files by uploading new files and removing the already existing, if required, which are stored in the Files Data store.		
Trigger: Evaluator needs to add or modify the assignment instructions and other relevant files.		
Type: External		
Preconditions: 1. The system should be up and running. 2. The Evaluator should be logged-in into the system with valid credentials. 3. The database is available and has previously stored files (if any).		
Normal Course: 1.0 Modifying the assignment instructions and other relevant files. 1. The evaluator specifies to upload new files or delete existing files. (Alternative Course 1.1) 2. The system navigates user to browse the files from the local disk. 3. User selects the required file from local disk to be uploaded 4. The system prompts the request whether to confirm or cancel the upload. (Alternative Course 1.2) 5. The evaluator confirms the upload request. 6. The system updates the files in the data store and lists in the 'Files' Section.		**Information for steps:** User selection List of files Files selection Upload Confirmation New Files
Alternative Course: 1.1 The evaluator chooses to delete files from the system. 1. The evaluator specifies/ selects the files which are required to be deleted. 2. The system prompts the confirmation request whether to proceed with deletion or not. (Alternative Course 2.1)		 List of files to be deleted Confirmation delete request

3. The evaluator confirms the deletion request.	Confirmation to delete
4. The system removes all the specified files from the database.	File to be deleted
1.2 Evaluator specifies not to proceed with the file upload. (Branch at step 3)	
1. The system asks the Evaluator whether to terminate upload.	Confirmation request to cancel upload
2a. The evaluator specifies to cancel the upload.	Confirmation to cancel upload
3a. The system terminates the use case.	
1.3 Evaluator specifies not to proceed with the file deletion. (Branch at step 7)	
1. The system asks the Evaluator whether to terminate deletion.	Confirmation request to cancel delete
2a. The evaluator specifies to cancel the deletion.	Confirmation to cancel delete
3a. The system terminates the use case.	

Post conditions:

1. The uploaded files are stored in the database.
2. The deleted files are removed from the database.
3. The 'Files' section is updated with the list of recently uploaded relevant files.

Exceptions:

E1. Invalid files with invalid file extensions are selected by the evaluator.

1. The system displays the message: "Please select valid files with valid file formats"
2. The system asks evaluator if he/ she wants to Re-select or cancel the upload process.

3a. The evaluator chooses to re-select the upload process.

4a. The evaluator re-specifies the files to be uploaded.

3b. The evaluator cancels the upload process.

4b. The system terminates the use case.

Summary			
Inputs	**Source**	**Output**	**Destination**
User selection	Evaluator	List of files	Evaluator
Files selection	Evaluator	New Files	Files datastore
Upload Confirmation	Evaluator	Confirmation delete request	Evaluator
List of files to be deleted	Evaluator	File to be deleted	Files datastore
Confirmation to delete	Evaluator	Confirmation request to cancel upload	Evaluator
Confirmation to cancel upload	Evaluator	Confirmation request cancel delete	Evaluator

9.2.7 USE CASE VII

Use Case Name: Submitting an assignment	**ID:** UC-7	**Priority:** High
Actor: Student		
Description: Student uploads the Financial statement report on Canvas system for his/her group		
Trigger: Student requests for assignment submission **Type:** External		
Preconditions: 1. The system is up and running. **2.** Student has logged in to the system with valid credentials and has navigated to assignment submission page. 3. The assignment instruction file should be uploaded on canvas for students.		
Normal Course: 1.0 Evaluating assignment 1. Student selects submit assignment option. 2. System displays option to select the files to be uploaded from student's local system. 3. Student uploads the assignment file. 4. System requests a submission confirmation from student. 5. Student confirms the submission. 6. System checks the assignment deadline and displays a message based on on-time or late submission. 7. System displays 'turned-in' message for on-time submission. 8. System displays an option to re-submit the assignment.		**Information for Steps** Submit request List of local files Files to be uploaded Confirmation request Confirmation Turned-in message Resubmit option
Alternate Course: (Branch at Step 6) 1. System displays 'late submission' message for post deadline submissions.		Late submission message
Postconditions: 1. Database is updated with the latest assignment file with submission related details. 2. Evaluator is notified on canvas for the submission made.		

Summary

Inputs	**Source**	**Outputs**	**Destination**
Submit request	Student	List of local files	Student
Files to be uploaded	Student	Confirmation request	Student
Confirmation	Student	Turned-in message	Student
		Resubmit option	Student
		Late submission message	Student

10. Context and Data Flow Diagrams

10.1 As- Is System

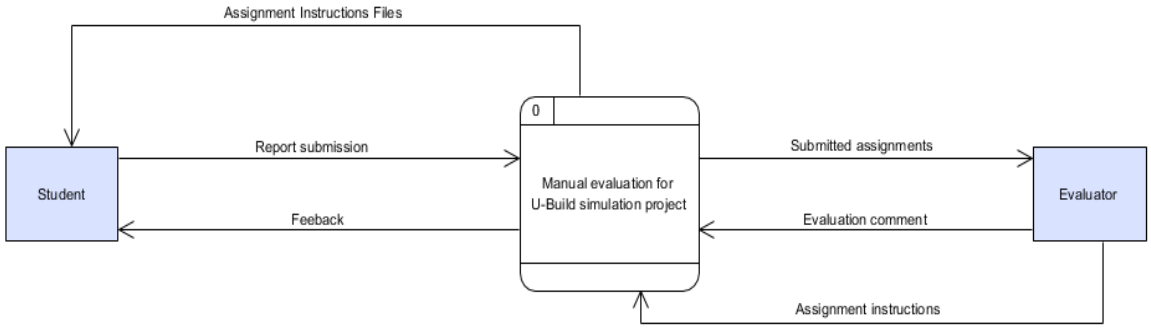


Illustration 8: Context diagram for As-Is System

Level 0

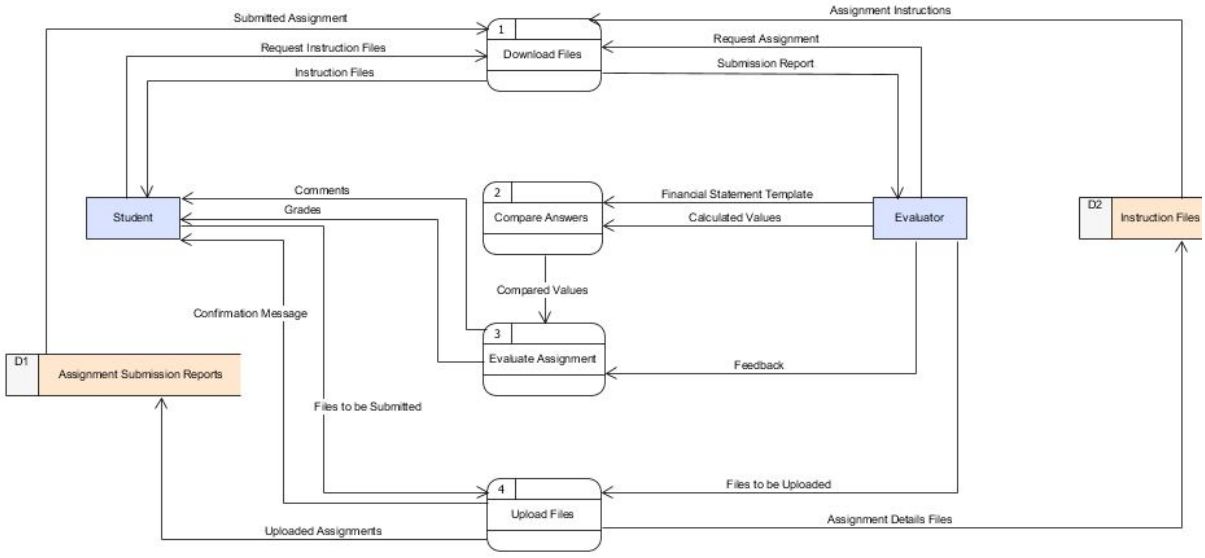


Illustration 9: Level 0 data flow diagram for As-Is System

10.2 To- Be System

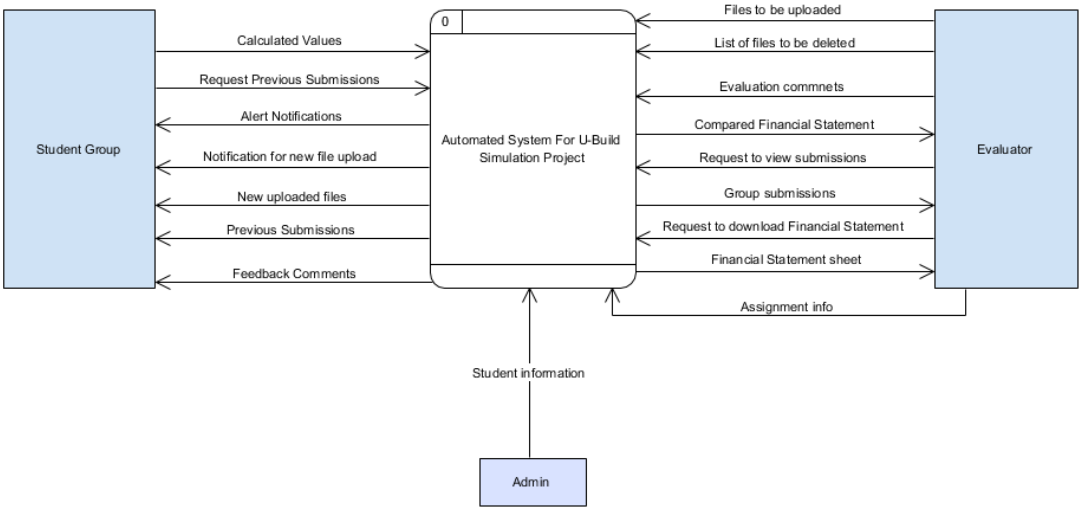


Illustration 10: Context Diagram for To- Be system

10.3 Data Flow Diagrams

Level 0

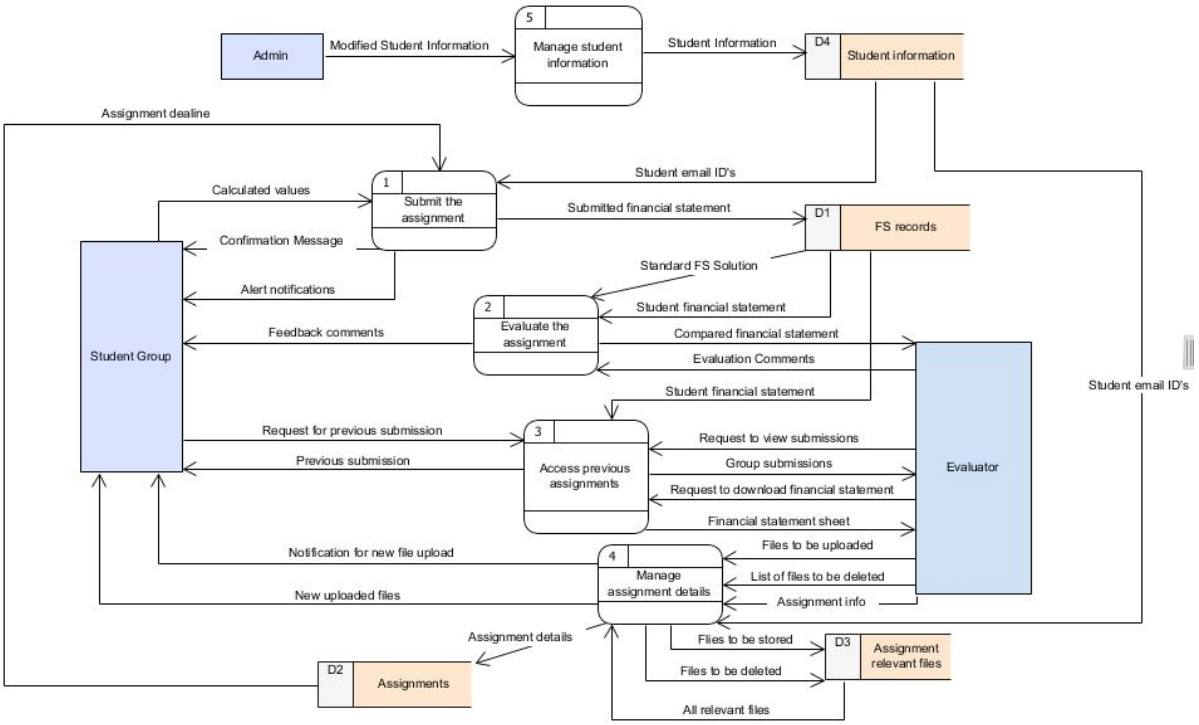


Illustration 11: Level 0 data flow diagram for To- Be system

Level 1

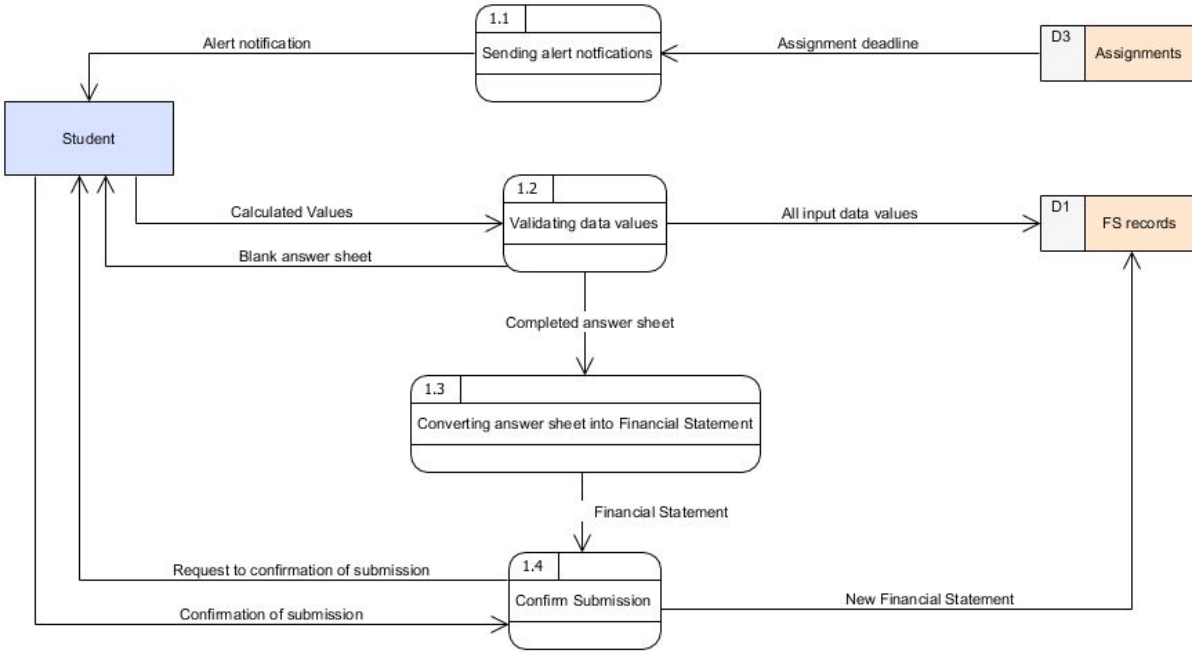


Illustration 12: Level 1 data flow diagram for Process 1 in To- Be system

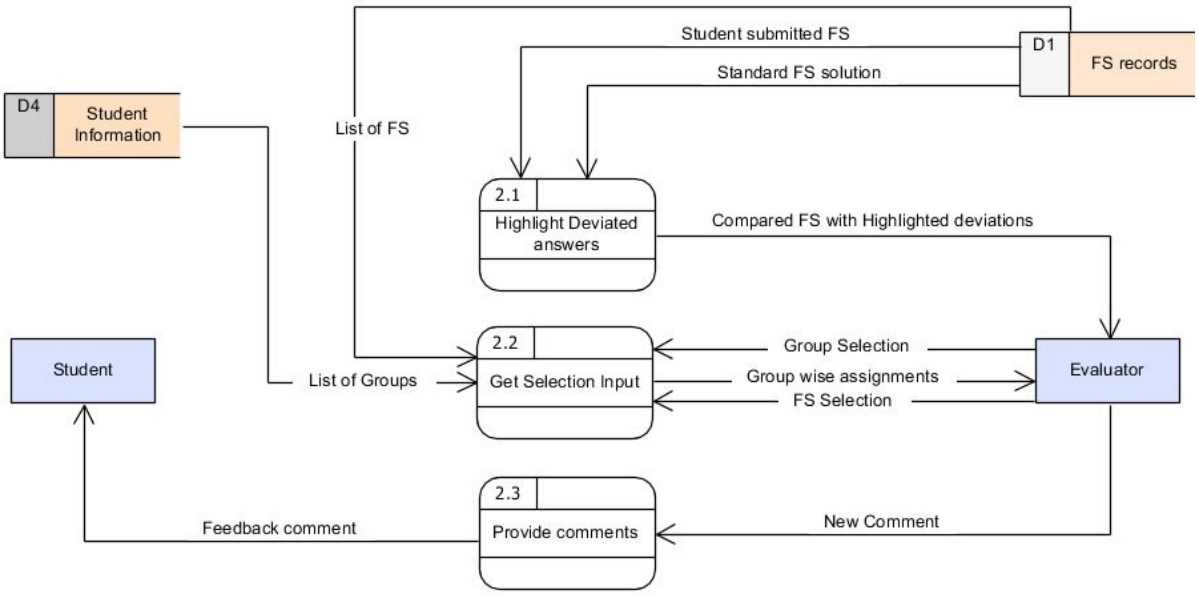


Illustration 13: Level 1 data flow diagram for Process 2 in To- Be system

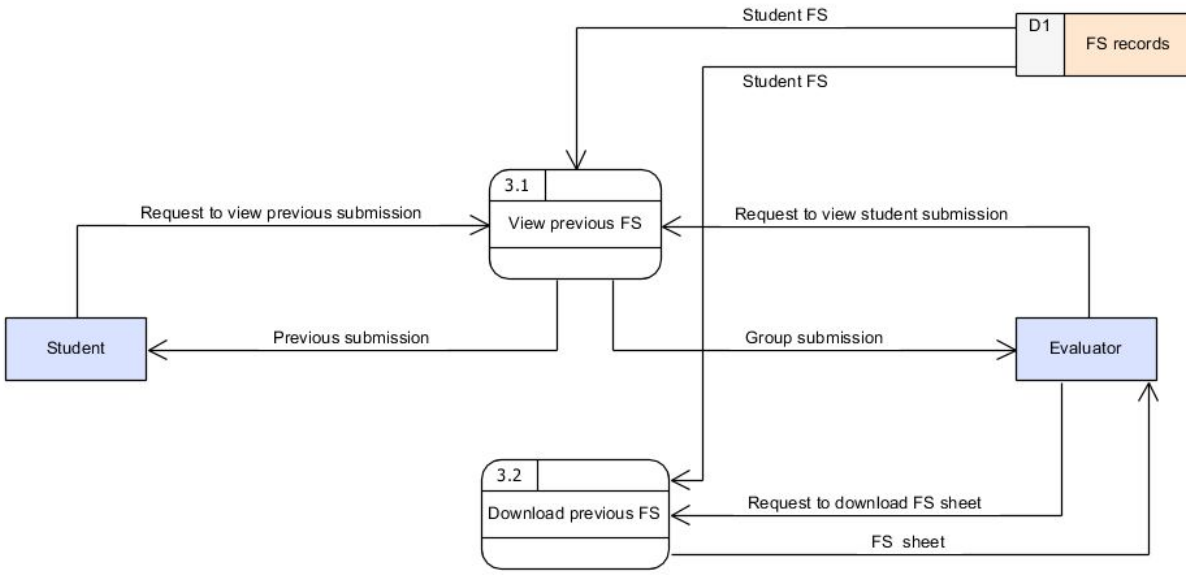


Illustration 14: Level 1 data flow diagram for Process 3 in To- Be system

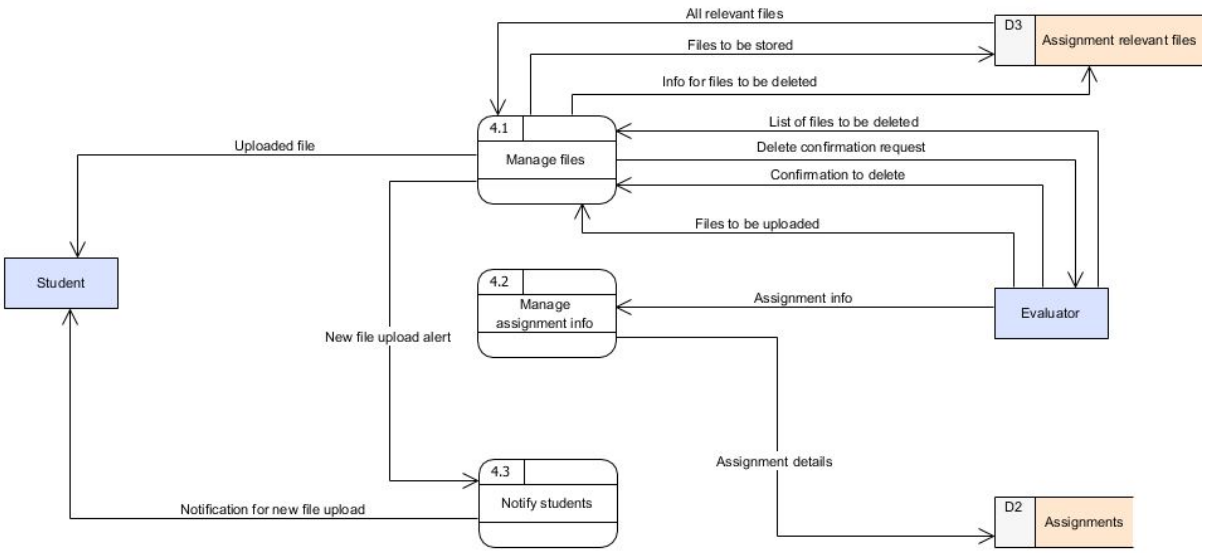


Illustration 15: Level 1 data flow diagram for Process 4 in To- Be system

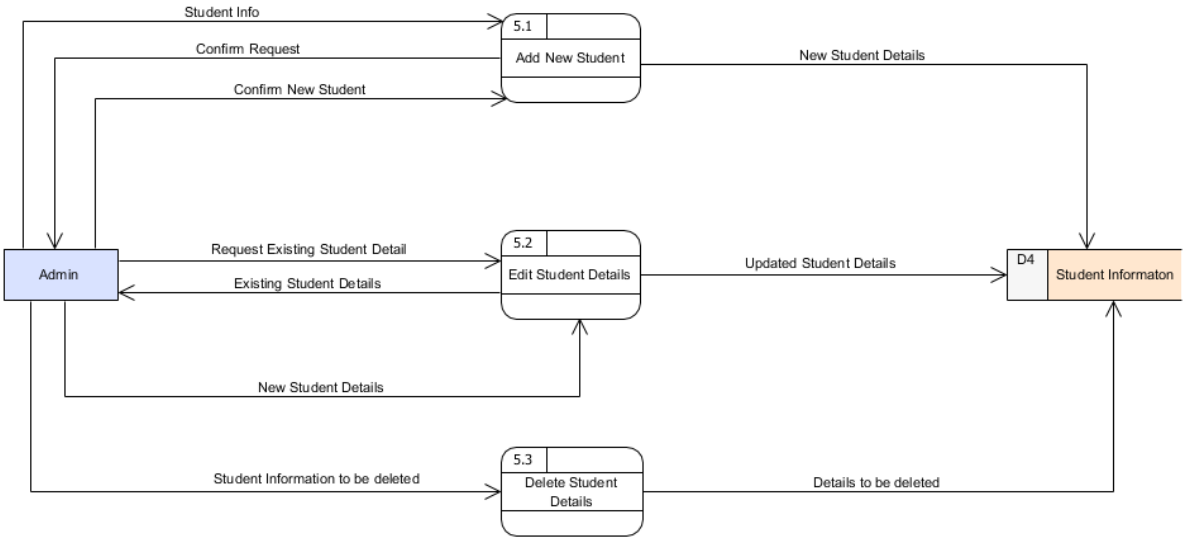


Illustration 16: Level 1 data flow diagram for Process 5 in To- Be system

Difference between the As-is and To-be system:

As-is system

In the current state of the system, the students are asked to calculate financial statements for the case of 'U-build' project. Students are divided into groups and proceed with the same groups for the entire project. Professor sends out 6 assignments to complete the project for which the professor has a standard solution and working methodology and the student groups are expected to submit their solutions for each of the assignments.

Currently, students submit the solutions, on Canvas, in the form of financial reports which comprises of final calculated values based on their own assumptions and some standard values given by professor.

The assignments are evaluated either by the professor or Teaching Assistant by accessing the assignment from Canvas, picking every value in the financial report and applying it to an Excel sheet to generate a financial statement and then testing it with the standard solution and methodology. Deviations from the expected values are figured out for grading the assignment. This is done for every assignment submitted by every group in the class.

To-be System

The to-be varies from the existing system in the way the students submit the assignments and professor/TA evaluates them.

A web application will be developed where the students will submit the financial reports and also enter the values of the calculations on a standard template form.

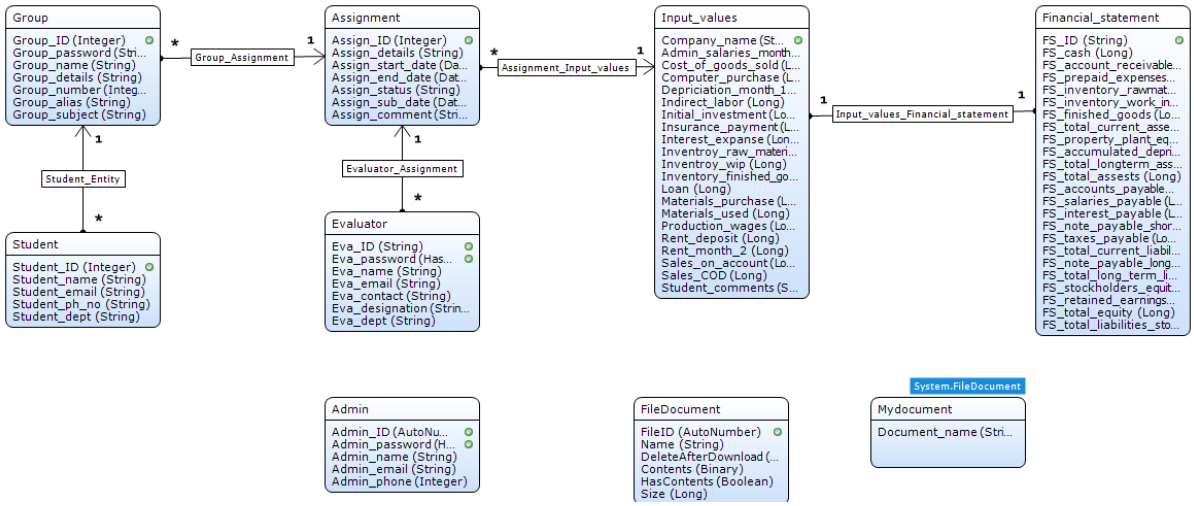
The system automatically converts it into a Financial statement, compares it with the standard solution, highlights the deviations and stores it. The professor can then directly

access the compared financial statement and check it to give feedback and comments for students' work.

It also has provision for students and evaluator to access the previously submitted assignments and to access other assignment related files.

11. Domain Model

11.1 Domain Model



11.2 Data Dictionary

All the data required by the application is stored in the specified domain model. There are nine entities in the model.

To implement three different home pages for the different users- Evaluator, Admin and Group, there are three entities which are as follows:

1) Student (**Std_ID**, Std_Name, Std_email, Std_ph_no, Std_dept)

- This entity stores details of the students such as their ID, Name, Contact Information and Department. The professor references this list of students at the very beginning of the course. Also, the groups are being formed by referencing the students from this list.

2) Group (**Group_ID, Group_password**, Group_name, Group_details, Group_number, Group_alias, Group_subject)

- Group entity stores the data required for the group login, i.e., ID, Password, Name and other details.

3) Evaluator (**Eva_ID, Eva_password**, Eva_name, Eva_email, Eva_contact, Eva_designation, Eva_dept)

- This entity stores the necessary data for the evaluator login, i.e., ID, Password, Name, Department, etc.

4) Admin (**Admin_ID, Admin_password**, Admin_name, Admin_email, Admin_phone)

- This entity stores the required details for the admin user login. This entity is not associated with other entities as this entity is created to perform future maintenance.

5) Assignment (**Assign_ID**, Assign_details, Assign_start_date, Assign_end_date, Assign_status, Assign_sub_date, Assign_comment)

- This entity stores the assignment related information.

6) Input_values (**Company_name**, Admin_salaries_month1, Cost_of_goods_sold, Computer_purchase, Depriciation_month_1, Indirect_labor, Initial_investment, Insurance_payment, Interest_expense, Inventory_raw_materials, Inventory_wip, Inventory_finished_goods, Loan, Materials_purchase, Materials_used, Production_wages, Rent_deposit, Rent_month_2, Sales_on_account, Sales_COD)

- This entity stores the data entered by the students in the specified form. The attributes of this entity are associated with the attributed present in the assignment memo form.

7) Financial_Statement (**FS_ID**, FS_cash, FS_account_receivable, FS_prepaid_expenses, FS_inventory_rawmaterials, FS_inventory_work_in_progress, FS_finishes_goods, FS_total_current_assests, FS_property_plant_equipment, FS_accumulated_depriciation, FS_total_longterm_assets, FS_total_assets, FS_accounts_payable, FS_salaries_payable, FS_interest_payable, FS_note_payable_short_term, FS_taxes_payable, FS_total_current_liabilities, FS_note_payable_long_term, FS_total_long_term_liabilities, FS_stockholders_equity, FS_retained_earnings, FS_total_equity, FS_total_liabilities_stockholders_equity)

- This entity stores the data associated with the created financial statements.

8) FileDocument (File_ID, Name, DeleteAfterDownload, Contents, HasContents, Size)

- This entity stores the details of static files uploaded by the users in the application.

9) MyDocument (Document_name)

- This Entity stores the actual static files

12. Mendix Project URL

https://mis573s18p-group1.mxapps.io/

User Stories Implemented

- ✓ As a student, I should be able to enter the values of my financial statement calculations so that the professor can evaluate them.
- ✓ As a user, I want the system to be running on a web application so that students and professor can access the system remotely.
- ✓ As an evaluator, I should be able to upload files relevant to assignments in the system so that, students can refer those files to complete their assignment.
- ✓ As a user, I want the web application to be supported and working on all web browsers, so that the system should be platform independent.

13. References

(https://www.glassdoor.com/Salaries/index.htm)
(https://www.merchantaccountsolutions.com/product)
(https://docs.oracle.com/cd/B25329_01/doc/license.102/b25456/toc.htm)
(http://www.cs.toronto.edu/~sme/CSC340F/readings/PIECES.html)
(Roth, 2013)
(Microsoft Project)
(draw.io)
(Visual Paradigm)

End of Document

Appendix C- Experience

Member 1 - Suraj Patil

As a systems analyst, it was challenging to understand what the client expects the system to give as the end output. One of the key lessons learnt is- the client requirements may vary or client may come up with newer ones initially and during the course of time. For instance, the client did not have a requirement on the look and feel of the financial statements being created in the initial design stage. However, in the final design phase, client gave a requirement to change the look and feel of the financial statements to be in-line with the excel copy of the financial statements that was provided to us for developing the system. Considering the time crunch and the available resources, it came as a surprise for me to implement this functionality. We managed to deliver the said functionality to the client before the project deadline. However, it was one of the significant challenges we faced while developing our application on Mendix. Had it been a different development application, it would have been easier to adjust the look and feel as per requirement. However, Mendix has the limitations that makes it difficult for developers to develop the system in a way client expects. Takeaway from this project I had is, always involve client in the development stages from the very first day so that application need not be changed in the later stages which is more difficult than making changes in the initial stages.

Member 2 - Manisha Chouhan

In our meeting with client post deliverable two presentation, we gave a demo of the system developed so far. As client viewed the application, she provided her valuable inputs and modifications to be made. One of the significant changes that were proposed as a part of the design in this meeting was, no standard financial statement would be common to all the group assignments. As and when students assume input values, the evaluator will apply the formulae and then evaluate and compare the financial statements. As a systems analyst, I understood that our understanding about client's requirement was incorrect and there is no standard financial statement solution to be compared and designed. The new system solely relied on the input values provided by users based on which calculations will be dependent.

Member 3 - Anuj Vyas

Developing a system using Mendix was challenging considering the restrictions the platform brings in, even though the software was easy to use. During my previous semester, I have used Android Studio to develop my application which was easier to learn and use simultaneously. Also, it provided flexibility to implement functionalities which are restricted to some extent in mendix. For instance, client expected the input values to be shown in a specific format in the file being downloaded. Mendix supports Excel format file download which couldn't serve client's purpose of financial statements format. Hence, as an analyst, it was challenging to come up with a workaround to suffice client's requirement using the available resources and technologies.

Member 4 - Vishakha Jadhav

Key lessons and takeaways from completing the project are to conduct more frequent review meetings with the project sponsor during the initial planning and analysis stages and to get the user requirements clearly understood at the beginning. In future, for any project, I would prioritize the tasks and revisit the project plan to use the resources efficiently whenever the client comes up with a new or modified requirement based on the prototype developed. I also realized the importance of having a team which has the right mix of analysts, developers and testers as the current team. I should also be prepared well enough to handle any of the surprises that might come on my way and have buffer time to manage it too so that the project is delivered as scheduled.

Member 5 - Nevetha Ramesh

The project was helpful to learn and realize key lessons required to design and develop a new system from scratch. One of the important lessons is that not to choose a complex or a huge project to complete in a short duration when the familiarity with the technology and the business process is considerably less. Mendix was new to the team and the scope of the project was huge which made it difficult to handle as the project continued. It came out to me as a surprise when some minor user stories, which were believed to be trivial and easy, that was kept to the end actually consumed a lot of time and effort to complete. In future, as a system analyst or a developer, one thing that I would improve on is performing a more number of black-box and white-box tests so that it becomes less complex in the later phases of testing.

Mendix URL

https://mis573s18p-group1.mxapps.io/index.html?profile=Responsive

Credentials

Role: Admin	ID: Admin	Password: taV2yMfu4G
Role: Evaluator	ID: Miller	Password: Miller123
Role: Student Group	ID: Group1	Password: Group123
Role: Student Group	ID: Group2	Password: Group234

References

https://docs.mendix.com/deployment/on-premises/design-the-architecture