|  |
| --- |
|  |
| Requirements Document |
| Team 6: Education++  Submitted: March, 21, 2013 |

|  |
| --- |
| Created By: |
| Husain Fazal |
| Preshoth Paramalingam |
| Robert Stanica |

Contents

1.1 The Business Problem 4

1.2 The Business 4

1.3 Problem Statement 4

2 Stakeholders 5

3 Business Requirements Overview 6

3.1 Incentives 6

3.1.1 Add Reward 6

3.1.2 Update Reward 6

3.1.3 Delete Reward 6

3.1.4 Add Badge 6

3.1.5 Update Badge 6

3.1.6 Delete Badge 6

3.1.7 View All Incentives (Badges and Rewards) 7

3.2 Point Earning Scenarios 8

3.2.1 Create Point Earning Scenario 8

3.2.2 Update a Point Earning Scenario 8

3.2.3 Delete a Point Earning Scenarios 8

3.2.4 View All Point Earning Scenarios 8

3.2.5 Gradebook Parser 8

3.3 Store 9

3.3.1 View all Available Rewards/Store Front 9

3.3.2 Purchase a Reward 9

3.3.3 View Points Balance 9

3.4 Leaderboard 10

3.4.1 View the Class Leaderboard 10

3.4.2 View the School Leaderboard 10

3.4.3 Student Profile 10

3.4.4 Opt-out of the Leaderboard 10

3.4.5 Opt-in to the Leaderboard 10

3.5 Notifications 11

3.5.1 View all new Notifications 11

3.5.2 View all Notifications 11

3.5.3 Send a Notification 11

3.6 Reporting 12

3.6.1 View your Transaction History 12

3.6.2 View a Student’s Transaction History 12

3.6.3 Track Reward Awarding Status 12

3.6.4 Generate a PDF for Track Reward Awarding Status 12

4 Business Rules 13

5 Use Cases 14

5.1 Use Case Diagram 14

5.2 Use Case: Manage Incentives 15

5.2.1 Business Rules That Apply 15

5.2.2 Scenario: Add a Reward 15

5.2.3 Scenario: Update a Reward 15

5.2.4 Scenario: Delete a Reward 15

5.2.5 Scenario: Add a Badge 16

5.2.6 Scenario: Update a Badge 16

5.2.7 Scenario: Delete a Badge 16

5.2.8 Scenario: View all Incentives (Regardless of Visibility) 16

5.3 Use Case: Manage Point Earning Scenarios 17

5.3.1 Business Rules That Apply 17

5.3.2 Scenario: Add a Point Earning Scenario 17

5.3.3 Scenario: Add a Point Earning Scenario (Conflict with other scenario) 17

5.3.4 Scenario: Update a Point Earning Scenario 18

5.3.5 Scenario: View all Point Earning Scenarios 18

5.3.6 Scenario: Delete a Point Earning Scenario 18

5.4 Use Case: Manage Store 19

5.4.1 Business Rules That Apply 19

5.4.2 Scenario: Display Currently Available Rewards 19

5.4.3 Scenario: Attempt to Purchase a Reward, Successful 19

5.4.4 Scenario: Attempt to Purchase a Reward, Unsuccessful (Inadequate Points) 19

5.5 Use Case: Manage Leaderboard 20

5.5.1 Business Rules That Apply 20

5.5.2 Scenario: View the Class Leaderboard 20

5.5.3 Scenario: View the School Leaderboard 20

5.5.4 Scenario: Opt-out of the Leaderboard System 20

5.5.5 Scenario: Opt-in to the Class Leaderboard 20

5.5.6 Scenario: View Student Profile 21

5.6 Use Case: Manage Notifications 22

5.6.1 Business Rules That Apply 22

5.6.2 Scenario: Send Students a Notification 22

5.6.3 Scenario: View all New Notifications 22

5.6.4 Scenario: View all Notifications 22

5.7 Use Case: Manage Reporting System 23

5.7.1 Business Rules That Apply 23

5.7.2 Scenario: View all Transactions made by a Student 23

5.7.3 Scenario: View all Transactions made by the User 23

5.7.4 Scenario: Generate a Report of Reward Purchases 23

5.7.5 Scenario: Generate a Report of Reward Purchases 23

5.7.6 Scenario: Edit the Awarded status of Reward Purchases 24

6 Operational Requirements 25

6.1 Security 25

6.2 Legal 25

7 Technical Requirements 26

8 Appendices 27

8.1 Appendix A: Individual Contributions 27

8.2 Appendix B: Log of Meetings 27

8.2.1 May 29th 2012 6:00PM, Office of Prof. Barb Czegel 27

8.2.2 June 14th 2012 6:00PM, Seneca College 27

8.2.3 July 19th 2012 6:30PM, Office of Prof. Mohamed Kassim 27

8.2.4 July 26th 2012 6:00PM, Seneca College 27

8.2.5 August 8th 2012 6:00PM, Office of Prof. Mohamed Kassim 28

The Business Problem

## The Business

Education++ is an organization that is aiming to incorporate elements of “gaming”, such as incentives and a rewards system, within the current education system used by universities and colleges. Education++ tracks student progress and achievements across all tests, quizzes and assignments and then offers rewards such as “Incentives” professors can offer to students that affect a student’s progress in the class.

Since Education++ is a start-up business, the business area we will be addressing is the actual Product Development business area. The Product Development business area within Education++ is responsible for tracking and handling classes and the “Reward” system that correlates with it. This business area is also responsible for the video and standard quizzing software components to allow professors to quiz students.

## Problem Statement

The issue that students across the world are facing, an issue they may not even know they have; it is the need for instant gratification. Thanks to the increase in popularity of video games and the advent of social media services like such as Facebook and Twitter, students have become conditioned to expect rewards for even their most basic efforts. This need for instant gratification has resulted in students failing to find motivation in school.  In today’s education system, getting a B+ on a test has no real immediate benefit. Scholarships provide incentive but they often have highly specific requirements and there’s no guarantee in actually getting one. Students stand to benefit exponentially when presented with the right motivation.

The software system that Education++ has envisioned will provide students the motivation they need to do the very best they can in school. By using rewards that can provide an instant and tangible incentive and badges that allow students to show off their accomplishments, Education++ will add a layer to the current educational system that will mimic the same elements that make video games and the aforementioned social media services addictive. The platform will allow students to keep track of their class activities, and see what incentives they’ve received. Professors will have the ability to create rewards and point earning scenarios for students to earn. It should be noted that the Education++ system is to be used as an incentive over a course’s current curriculum. That is, the breakdown of course marks cannot include Education++ rewards or components; they are a bonus.

# Stakeholders

|  |  |
| --- | --- |
| **Stakeholder** | **Role** |
| Education++: Husain Fazal Preshoth Paramalingam Robert Stanica | Sponsor & Authority |
| Barb Czegel Mohamed Kassim | Authority |
| Students | End User |
| Professor | End User |
| Colleges/Universities | Customer/Onlooker/Advisors |

# Business Requirements Overview

The proposed system includes an Incentives component, a Point Earning Scenarios component, a Store component, a Leaderboard component, a Notifications component and a Reporting component.

## Incentives

This part of the software will allow professors to add, update, delete or view all incentives (badges or rewards). The purpose of this component is for Professors to create “inventory” for students within a class to purchase from the store.

### Add Reward

End user: Professor

* Professor will enter a reward name, description, point value, store-visibility, quantity-per-student, expiry date and an icon selection from a list of options.
* The professor will confirm the entry and varying on the store-visibility, the item will be available in the store

### Update Reward

End user: Professor

* Professor will select the reward they wish to edit
* The system will return all details for that reward and the Professor is presented with the ability to change fields

### Delete Reward

End user: Professor

* Professor will select the reward they wish to delete
* Professor confirms the removal of the class and the Reward is “hidden” but kept for transaction purposes View all Rewards

### Add Badge

End user: Professor

* Professor will enter a badge name, point value, store-visibility and an icon selection from a list of options.
* The professor will confirm the entry and varying on the store-visibility, the item will be available in the store

### Update Badge

End user: Professor

* Professor will select the reward they wish to edit
* The system will return all details for that reward and the Professor is presented with the ability to change fields

### Delete Badge

End user: Professor

* Professor will select the reward they wish to delete
* Professor confirms the removal of the class and the Reward is “hidden” but kept for transaction purposes View all Rewards

### View All Incentives (Badges and Rewards)

End user: Professor

* Professor selects to view all rewards and the system displays all rewards along with their individual details (Name, Expiry Date – if necessary, price, description)

## Point Earning Scenarios

This part of the software will allow professors to add, update, delete or view all point earning scenarios (scenarios in which students can earn points). The purpose of this component is for Professors to create opportunities for students within a class to earn points.

### Create Point Earning Scenario

End user: Professor

* Professor will enter a scenario name, description, point-value, and expiry date.
* Then given a list of activities available in the grade book, the Professor will select activities, their condition (greater than, equal, less than), and a point value (100 for complete, 0 for incomplete, or a percent value)
* The professor will confirm the entry and if the requirements are not the exact same as another scenario, the point earning scenario is activated

### Update a Point Earning Scenario

End user: Professor

* Professor will select the Point Earning Scenario they wish to edit
* The system will return all details for that Point Earning Scenario and the Professor is presented with the ability to change fields
* The professor will confirm the update and if the requirements are not the exact same as another scenario, the point earning scenario is activated

### Delete a Point Earning Scenarios

End user: Professor

* Professor selects a Point Earning Scenario from the list of current Point Earning Scenario and is asked to confirm removal
* Professor confirms and the Point Earning Scenario is “hidden” but kept for history purposes

### View All Point Earning Scenarios

End user: Professor

* Professor selects to display all Point Earning Scenarios and a list of all Point Earning Scenarios is displayed

### Gradebook Parser

End user: Professor, Student

* Determines if a PES’ requirements have been met
* Automatically triggered

## Store

This part of the software will allow professors and students to view all incentives (badges and rewards) and for students to purchase them (if they meet the requirements).

### View all Available Rewards/Store Front

End user: Professor, Student

* Student or Professor enters the store and is greeted by all visible and not “hidden” items
* Professor’s buy button disabled

### Purchase a Reward

End user: Student

* Student will select the purchase button next to a reward
* The System will assess whether or not the student has adequate quantity and points for the purchase (Checking for a Fake Incentive ID, Correct Course, Store Visibility, Deleted Status, Expired Status, Quantity Status [per student], and point balance)
* The transaction is logged, and a record of the purchase is made

### View Points Balance

End user: Student

* Upon opening the store front, the student’s point balance is displayed

## Leaderboard

This part of the software will create a competitive element between classmates and peers in the school, allowing students to view a “leaderboard” of accumulated point-balances, and badges. There is an option to opt-out of the leaderboard.

### View the Class Leaderboard

End user: Professor, Student

* Professor or Student selects the option to view the Class Leaderboard
* All opted-in students are displayed with their accumulated points-balance for that course in that descending order.
* Columns are sortable by Name or Accumulated Points
* The student’s profile picture is also displayed
* The student who requested to view the leaderboard is highlighted

### View the School Leaderboard

End user: Professor, Student

* Professor or Student selects the option to view the School Leaderboard
* All opted-in students are displayed with the 3 most recently earned badges as well as their total badge count for all Education++ courses in that descending order.
* Columns are sortable by Name or Badge Count
* The student who requested to view the leaderboard is highlighted

### Student Profile

End user: Professor, Student

* Actor clicks on a Student Name, a view is displayed with the Student’s profile picture, name, badge count, badge icons and enrolled Education++ courses.
* Students opted out of the leaderboard do not have publically visible profiles

### Opt-out of the Leaderboard

End user: Student

* Student selects to opt-out of the leaderboard and confirms
* The student is marked as opted out of the leaderboard

### Opt-in to the Leaderboard

End user: Student

* Student selects to opt-in to the leaderboard and confirms
* The student is marked as opted in to the leaderboard

## Notifications

This part of the software will allow for professors to display all purchases made in the class, view all student’s point balances, actually triggering all Point Earning Scenarios and Generates reports for new purchases for Professors.

### View all new Notifications

End user: Student

* Student selects to view all new notifications (main page)
* All new notifications (not dismissed) are displayed

### View all Notifications

End user: Student

* Student selects to view all notifications
* All notifications are displayed, ordered by newness

### Send a Notification

End user: Professor

* Professor selects to send a notification to all students enrolled in the class
* The professor enters a title and the message and hits save

-

## Reporting

This part of the software will allow for professors and students to track transactions including earned points and spent points (points spent in the store on rewards and badges).

### View your Transaction History

End user: Student

* Student selects to view their own transaction history
* A list of all transactions is displayed including store purchases and point earning scenarios that have been met, organized by date

### View a Student’s Transaction History

End user: Professor

* Professor selects to view a student’s transaction history
* Professor selects a student from the dropdown menu
* A list of all transactions is displayed including store purchases and point earning scenarios that have been met, organized by date

### Track Reward Awarding Status

End user: Professor

* Professor selects to view a report of reward purchase tracking
* Professor can update awarded rewards for students

### Generate a PDF for Track Reward Awarding Status

End user: Professor

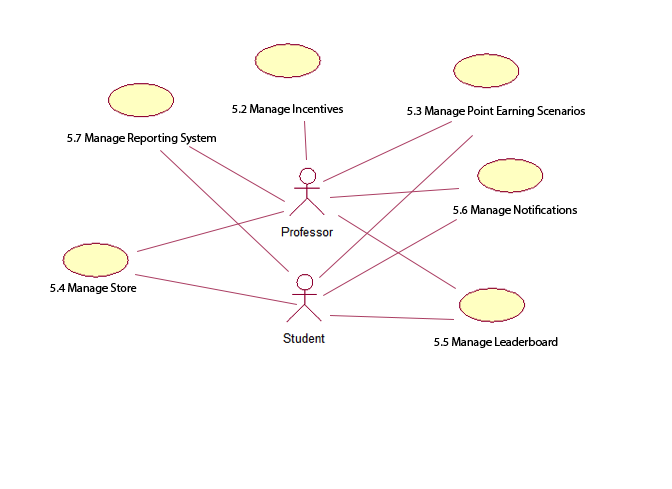
* Professor selects to view a report of reward purchase tracking
* Selects generate a PDF

# Business Rules

|  |  |
| --- | --- |
|  | Description |
|  | The Education++ system is to be used as an incentive over a course’s current curriculum. That is, the breakdown of course marks cannot include Education++ rewards or components; they are a bonus. |
|  | A point earning scenario cannot have the same requirements as another |
|  | Students are set as “opted-in” initially for the leaderboard |
|  | Students can opt-out or opt-in of the leaderboard at anytime |
|  | Only students can purchase rewards |
|  | Only a student with adequate points and quantity can make a purchase |
|  | Only professors can view the Rewarding Report system |
|  | Notifications will automatically be purged from the database 90 days after creation |
|  | A reward cannot have the same name as another reward |
|  | Rewards and Badges MUST have images, if not a predefined image is set |
|  | Each student may only purchase one badge |

# Use Cases

## Use Case Diagram



Use Case: Manage Incentives

This use case allows a professor to create, update, view or delete a reward or badge within a specific class.

### Business Rules That Apply

* Incentives cannot have the same name as another incentive
* Rewards and Badges MUST have images, if not a predefined image is set

### Scenario: Add a Reward

Preconditions: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Professor) | System |
| Requests to create a new Reward | Returns a form input for reward name, description, points value, store-visibility, quantity per student, icon & expiry date |
| Enters reward name, description, points value, store-visibility, quantity-per-student, expiry date and selects an icon | Returns a confirmation screen |
| Confirms | Saves the new reward in the database |

### Scenario: Update a Reward

Preconditions: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Professor) | System |
| Requests to update a reward | Returns a prepopulated editable form input with reward name, description, points value, store-visibility, quantity per student, icon & expiry date |
| Updates reward name, description, points value, store-visibility, quantity-per-student, expiry date and updates icon selection | Returns a confirmation screen |
| Confirms | Saves the updated reward into the database |

### Scenario: Delete a Reward

Preconditions: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Professor) | System |
| Requests to view all rewards for the class | Returns a list of all rewards |
| Selects a reward on the list and requests to delete | Returns a confirmation screen |
| Confirms | Reward is marked as hidden |

### Scenario: Add a Badge

Preconditions: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Professor) | System |
| Requests to create a new Badge | Returns a form input for badge name, points value, store-visibility, & icon |
| Enters badge name, points value, store-visibility, expiry date and selects an icon | Returns a confirmation screen |
| Confirms | Saves the new reward in the database |

### Scenario: Update a Badge

Preconditions: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Professor) | System |
| Requests to update a badge | Returns a prepopulated editable form input with badge name, points value, store-visibility, & icon |
| Updates badge name, points value, store-visibility, and updates icon selection if neccessary | Returns a confirmation screen |
| Confirms | Saves the updated reward into the database |

### Scenario: Delete a Badge

Preconditions: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Professor) | System |
| Requests to view all rewards for the class | Returns a list of all rewards |
| Selects a reward on the list and requests to delete | Returns a confirmation screen |
| Confirms | Reward is marked as hidden |

### Scenario: View all Incentives (Regardless of Visibility)

Preconditions: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Professor) | System |
| Requests to view all incentives | Returns a list of information about all incentives including (for each) a name, description\*, points value, store-visibility, quantity-per-student, expiry date\* and icon selection (\* does not apply to badges) |

Use Case: Manage Point Earning Scenarios

This use case allows a professor to create, view, update or delete students (a) Manage Point Earning Scenario(s)

### Business Rules That Apply

* A point earning scenario cannot have the same requirements as another
* Expiry date must be after the date of the addition/editing of the Point Earning Scenario

### Scenario: Add a Point Earning Scenario

Precondition: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Professor) | System |
| Requests to add a new Point Earning Scenario | Returns a form with fields for: name, point-value, description requirements and expiry date. |
| Enters in name (of scenario), point-value, description requirements and expiry date. | Returns a list of activities. |
| Selects an activity, a condition (greater than, equal to, less than) and a value (100 for complete, 0 for incomplete and percent otherwise) | Saves requirement to database, System suggests a points-value |
| Loops until complete |  |
| Confirms to submit form | System checks if the current requirements are identical to another Point Earning Scenario. Since not, Saves Point Earning Scenario to database |

### Scenario: Add a Point Earning Scenario (Conflict with other scenario)

Precondition: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Professor) | System |
| Requests to add a new Point Earning Scenario | Returns a form with fields for: name, point-value, description requirements and expiry date. |
| Enters in name (of scenario), point-value, description requirements and expiry date. | Returns a list of activities. |
| Selects an activity, a condition (greater than, equal to, less than) and a value (100 for complete, 0 for incomplete and percent otherwise) | Saves requirement to database |
| Loops until complete | System suggests a points-value |
| Confirms to submit form | System checks if the current requirements are identical to another Point Earning Scenario. Scenario is found, informs user to change requirements. |
| Updates Requirements | System checks if the current requirements are identical to another Point Earning Scenario. Since not, Saves Point Earning Scenario to database |

### Scenario: Update a Point Earning Scenario

Precondition: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Professor) | System |
| Requests to update an existing Point Earning Scenario | Returns a list of existing Point Earning Scenario names |
| Selects a Point Earning Scenario name | Returns a form with prepopulated fields for: name, point-value, description requirements and expiry date. |
| Enters in name (of scenario), point-value, description requirements and expiry date. | Returns a list of activities as well as prepopulated activities (for existing requirements). |
| Selects an activity, a condition (greater than, equal to, less than) and a value (100 for complete, 0 for incomplete and percent otherwise) | Saves requirement to database, System suggests a points-value |
| Loop until complete |  |
| Confirms to submit form | System checks if the current requirements are identical to another Point Earning Scenario. Since not, Saves Point Earning Scenario to database |

### Scenario: View all Point Earning Scenarios

Precondition: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Professor) | System |
| Requests to view all existing Point Earning Scenarios | Returns a report with all of the Point Earning Scenarios’ along with their names, point-values, description requirements, expiry dates and a list of requirements (Activity, condition and value) each. |

### Scenario: Delete a Point Earning Scenario

Precondition: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Professor) | System |
| Requests to delete an existing Point Earning Scenario | Returns a list of existing Point Earning Scenario names |
| Selects a Point Earning Scenario name | Returns a report with the Point Earning Scenario’s name, point-value, description requirements, expiry date and a list of requirements (Activity, condition and value), and a request to confirm deletion |
| Confirms | Point Earning Scenario is marked as hidden |

Use Case: Manage Store

This use case allows a student to make purchases of rewards based on points they have earned.

### Business Rules That Apply

* Only students can purchase rewards
* Only a student with adequate points and quantity can make a purchase

### Scenario: Display Currently Available Rewards

Precondition: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Professor, Student) | System |
| Requests to view currently available rewards via the store front | Retrieves and displays a list of rewards (whose visibility is true) as well as the logged in user’s remaining quantity. |

### Scenario: Attempt to Purchase a Reward, Successful

Preconditions: Actor has been authenticated and is in the plug-in within the appropriate class and has selected a reward to purchase (From a list of currently available awards)

|  |  |
| --- | --- |
| Actor (Student) | System |
| Requests to purchase the reward, enters a quantity | The system asks to confirm the purchase. |
| Confirms | System validates the purchase: checks for a correct incentive ID, checks if the incentive is for the correct course, checks if the incentive is visible, checks if the deleted status is not set, checks if the incentive is not expired, checks if there is enough quantity available and checks if the point balance is valid. A transaction item is logged with the student, reward and date/time, a record of the Student purchasing the Incentive is made, the student’s point balance and quantity is updated and a notification is sent to the student. |

### Scenario: Attempt to Purchase a Reward, Unsuccessful (Inadequate Points)

Preconditions: Actor has been authenticated and is in the plug-in within the appropriate class and has selected a reward to purchase (From a list of currently available awards)

|  |  |
| --- | --- |
| Actor (Student) | System |
| Requests to purchase the reward, enters a quantity | The system asks to confirm the purchase. |
| Confirms | System validates the purchase: checks for a correct incentive ID, checks if the incentive is for the correct course, checks if the incentive is visible, checks if the deleted status is not set, checks if the incentive is not expired, checks if there is enough quantity available and checks if the point balance is valid. The student is told that the purchase was unsuccessful. |

## Use Case: Manage Leaderboard

This use case allows students and professors to view the leaderboard and to opt in or out of the leaderboard system

### Business Rules That Apply

* Students are set as “opted-in” initially
* Students can opt-out or opt-in of the leaderboard at anytime

### Scenario: View the Class Leaderboard

Precondition: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Student/Professor) | System |
| Requests to view the leaderboard | Returns a list of all student names (from the class) with their accumulated points balance (ordered by) and profile picture for those students who are opted in. |

### Scenario: View the School Leaderboard

Precondition: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Student/Professor) | System |
| Requests to view the leaderboard | Returns a list of all student names (from the school) with three of their badges and their badge count (ordered by) for those students who are opted in. |

### Scenario: Opt-out of the Leaderboard System

Precondition: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Student) | System |
| Requests to change the privacy setting | Returns student’s current privacy setting (opted in) |
| Updates to opt-out | Asks student to verify |
| Confirms | Student’s status is changed to opted-out |

### Scenario: Opt-in to the Class Leaderboard

Precondition: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Student) | System |
| Requests to change the privacy setting | Returns student’s current privacy setting (opted out) |
| Updates to opt-in | Asks student to verify |
| Confirms | Student’s status is changed to opted-in |

### Scenario: View Student Profile

Precondition: Actor has been authenticated and is in the plug-in within the appropriate class, has selected to view a leaderboard

|  |  |
| --- | --- |
| Actor (Student, Professor) | System |
| Requests to view a Student’s Profile | Returns student’s badges, name and profile picture |

## Use Case: Manage Notifications

This use case allows a professor or the system to generate and allow students to receive notifications.

### Business Rules That Apply

* Notifications will automatically be purged from the database 90 days after creation

### Scenario: Send Students a Notification

Precondition: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Professor) | System |
| Selects “Send Notification” | System returns a form with a content box |
| Enters the content/message and clicks send | System requests a confirmation |
| Confirms | System saves notifications into database |

### Scenario: View all New Notifications

Precondition: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Student) | System |
| Enters Education++ Environment | System returns a list of non-dismissed notifications |

### Scenario: View all Notifications

Precondition: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Student) | System |
| Student selects to view all Notifications | System returns a list of non-dismissed notifications followed by all dismissed notifications that are not older than 90 days |

## Use Case: Manage Reporting System

This use case allows a professor to view all purchases made by students, view all student’s point balances and track reward awarding’s

### Business Rules That Apply

* Only professors can manage the Rewarding Report System

### Scenario: View all Transactions made by a Student

Precondition: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Professor) | System |
| Requests to receive a student’s transactions | Returns a Select field of students |
| Selects a student | Retrieves and displays a report with a list of transactions purchases of incentives and earning of Point Earning Scenarios made by the selected student |

### Scenario: View all Transactions made by the Current User

Precondition: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Student) | System |
| Requests to receive all transactions made | Retrieves and displays a report with a list of transactions purchases of incentives and earning of Point Earning Scenarios made by the logged in student |

### Scenario: Generate a Report of Reward Purchases

Precondition: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Professor) | System |
| Selects the “Generate a Report Reward Purchases” | System reads through each purchased reward transaction and displays the list of transactions as well as a checkbox for each for the awarded status and a generate PDF button. |

### Scenario: Generate a Report of Reward Purchases

Precondition: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Professor) | System |
| Selects the “Generate a Report Reward Purchases” | System reads through each purchased reward transaction and displays the list of transactions as well as a checkbox for each for the awarded status, and a generate PDF button. |
| Professor Presses the Generate PDF button | System generates a PDF with the transaction and rewarding history |

### Scenario: Edit the Awarded status of Reward Purchases

Precondition: Actor has been authenticated and is in the plug-in within the appropriate class

|  |  |
| --- | --- |
| Actor (Professor) | System |
| Selects the “Generate a Report Reward Purchases” | System reads through each purchased reward transaction and displays the list of transactions as well as a checkbox for each for the awarded status. |
| Professor changes checkboxes and clicks save | System persists data |

# Operational Requirements

## Security

Based upon Moodle’s Security and Privacy policies

## Legal

Our customers have the right to privacy:

* Students will have an option whether or not to make their accumulated point balance visible for classmates to see ie. The “leaderboard” available for classmate’s viewing
* All student and professor data will be secured through Moodle
* All data pertaining to students and professors will not be distributed to any person or organization, except with the associated school they are apart of.

# Technical Requirements

Planned technology is as follows:

* Apache
* MySQL
* PHP
* Moodle
* Web-based
* JQuery

Reasons:

* These technologies are the backing behind Moodle, the system we will be creating a plug-in for.

# Appendices

## Appendix A: Individual Contributions

|  |  |
| --- | --- |
| **Name** | **Contributions** |
| Husain Fazal | * Took part in meetings to discuss functionality * Created “The Business” write up * Created the “Business Requirements Overview” * Logged meetings and wrote the Appendices * Final Report Assembly |
| Preshoth Paramalingam | * Took part in meetings to discuss functionality * Created Use Case Diagram * Created Use Cases 5.2-5.3 * Added to “Business Requirements Overview” * Created the “Business Rules” * Created the “Operational Requirements” |
| Robert Stanica | * Took part in meetings to discuss functionality * Created the “Problem Statement” write up * Created Use Cases 5.4-5.7 * Created the “Technical Requirements” |

## Appendix B: Log of Meetings

### May 29th 2012 6:00PM, Office of Prof. Barb Czegel

**In attendance: Husain Fazal, Preshoth Paramalingam, Robert Stanica, Barb Czegel**

* Discussed base idea behind Education++, why instant gratification is important
* Expanded the Trophy system into two parts:
  1. Incentives – Professors can use this to give students rewards such as an extra 5 minutes on an exam, a bonus percent etc.
  2. Badges – Professors can set achievements/milestones for students to achieve with a leaderboard system corresponding with Badges
* Broke down the Requirements Document into sections to complete

### June 14th 2012 6:00PM, Seneca College

**In attendance: Husain Fazal, Preshoth Paramalingam, Robert Stanica**

* Reviewed discussion from May 29th
* Assigned pieces of the Requirements Document to complete for a mid-July compilation deadline

### July 19th 2012 6:30PM, Office of Prof. Mohamed Kassim

**In attendance: Husain Fazal, Preshoth Paramalingam, Mohamed Kassim**

* Explained our refined version of the system to Professor Kassim
* Established our system as a class management/incentive system with two main parts: “Rewards” (Which consists of Incentives and Badges) and “Quizzes” (Which consists of Video and Standard Quizzes)

### July 26th 2012 6:00PM, Seneca College

**In attendance: Husain Fazal, Preshoth Paramalingam, Robert Stanica**

* Revision of the first compiled report
* Reviewed changes and established need for one more meeting with Professor Kassim to finalize

### August 8th 2012 6:00PM, Office of Prof. Mohamed Kassim

**In attendance: Husain Fazal, Preshoth Paramalingam, Mohamed Kassim**

* Major Draft was presented to Prof. Kassim
* Discussed final details to be included in Report. Changes:
  1. Add more details for Use Cases
  2. Add more details for Business Requirement Overview
  3. Ensure there “Add”, “Remove”, “View”, “Update” functionality for each data table
  4. Add summary information in Business Requirement Overview headers
  5. Add more Business Rules