|  |
| --- |
| *Use Case Name:* Register |
| *Participating Actors:* Guest, Email Server |
| *Pre-Condition:*   * Guest is not logged in. |
| *Flow of Events:*   1. Guest clicks ‘Register on the website. 2. EasyUni responds by presenting a register form to the user. *The form includes email and password field for the user to complete.* 3. Guest complete the register form with their email address and password. 4. EasyUni verifies that the email address entered has not been used before. 5. Email Server sends out a verification email to the email address provided. |
| *Post-Condition:*   * Guest entered an invalid email address. * Guest successfully registered. |
| *Alternate flows:*  AF-S4: EasyUni detects that the email address has already been registered as a user   1. EasyUni displays the message “Email address already in use!” for 2 seconds 2. Return to Step 2 |
| *Exceptions:*  Ex 1: Email Server fails to send the verification email due to server/network issues   1. EasyUni displays the message “Error in sending verification email!” for 2 seconds 2. Return to Step 2   Ex 2: System failed to capture the details into database   1. Prompt user the error “something went wrong. Please try again later!” 2. Return to Step 2. |

|  |
| --- |
| *Use Case Name:* Login |
| *Participating Actors:* Guest |
| *Pre-Condition:*   * Guest is not logged in. |
| *Flow of events:*   1. Guest clicks ‘Login’ on the website. 2. EasyUni responds by presenting a login form to the user. *The form includes email and password field for the user to complete.* 3. Guest complete the login form with their email address and password. 4. EasyUni authenticate if the email address and password matches the account details 5. Guest successfully logged in to their account. |
| *Post-Condition:*   * Guest entered an invalid email address. * Guest entered an invalid password. * Guest successfully logged in. |
| *Alternate flows:*  AF-S4: EasyUni detects that the email address used is not registered as a user   1. EasyUni displays the message “Invalid user!” for 2 seconds 2. Return to Step 2   AF-S4: EasyUni detects that the input password does not match the one in the database   1. EasyUni displays the message “Invalid password!” for 2 seconds 2. Return to Step 2 |
| *Exceptions:*  Ex 1: User has failed to login via the same email 3 times in a row within the span of 5 minutes   1. EasyUni times out the associated account under that email - access is locked even with the correct password 2. Email Server sends a Password Reset email to the email account |

|  |
| --- |
| *Use Case Name:* Reset Password |
| *Participating Actors:* Guest, Email Server |
| *Pre-Condition:*   * Guest is not logged in. |
| *Flow of events:*   1. Guest clicks ‘Reset Password’ on the website. 2. EasyUni responds by presenting a reset password form to the user. *The form includes email field for the user to complete.* 3. Guest complete the reset password form with their email address. 4. EasyUni verifies if the email address exists in the system. 5. Email Server sends out an email with detailed instructions on how to reset their password to the email address provided. |
| *Post-Condition:*   * Guest entered an invalid email address. * Email Server successfully sent out the email. |
| *Alternate flows:*  AF-S4: Invaild email address indicated in the reset password form   1. EasyUni Displays the message ” invalid email ! ” to the user for 2 seconds. 2. Return to step 3. |
| *Exceptions:*  *Ex 1 :The user fails to complete the reset password form in one hour.*   1. *Reset password form will become invalid, will not reset password.* 2. *Users will have to request another reset password form from EasyUni.* |

|  |
| --- |
| *Use Case Name:* Search Courses |
| *Participating Actors:* Guest |
| *Pre-Condition:* |
| *Flow of events:*   1. Guest type their search query onto the search bar on the website. 2. Guest click on the search button. 3. EasyUni responds by presenting a list of courses to the user matching to the search query. *Each of the list items includes course name and school name.* |
| *Post-Condition:*   * Search results have successfully displayed. |
| *Alternate flows:*  AF-S4: The guest types into the search query a course that is not found in any university   1. EasyUni will display “No such courses found!” 2. Return to step 1 |
| *Exceptions:* |

|  |
| --- |
| *Use Case Name:* View Courses |
| *Participating Actors:* Guest, GOV.SG API |
| *Pre-Condition:* |
| *Flow of events:*   1. Guest clicks ‘View Courses’ on the website. 2. EasyUni request salary data from GOV.SG API. 3. GOV.SG API responds by providing salary data for all courses. 4. EasyUni responds by presenting a list of courses to the user. *Each of the list items includes course name, IGP, salary and school name.* |
| *Post-Condition:*   * All courses in the system have been successfully displayed. |
| *Alternate flows:* |
| *Exceptions:*  Ex 1: GOV.SG API fails to respond   1. EasyUni will display “salary not found” on the salary data column instead. |

|  |
| --- |
| *Use Case Name:* Sort by IGP |
| *Participating Actors:* Guest |
| *Pre-Condition:*   * Guest must be on the ‘View Courses’ page |
| *Flow of events:*   1. Guest selects ‘Sort by IGP’ on the view courses page. 2. EasyUni responds by sorting all list items according to their IGP and displaying it to the user. |
| *Post-Condition:*   * All courses in the system have been successfully displayed and are sorted according to their IGP. |
| *Alternate flows:*  AF-S1: Sorted by Descending IGP   1. List items will be sorted from descending order according to their IGP |
| *Exceptions:* |

|  |
| --- |
| *Use Case Name:* Sort by Salary |
| *Participating Actors:* Guest, GOV.SG API |
| *Pre-Condition:*   * Guest must be on the ‘View Courses’ page |
| *Flow of events:*   1. Guest selects ‘Sort by Salary’ on the view courses page. 2. EasyUni request salary data from GOV.SG API 3. GOV.SG API responds by providing salary data for all courses. 4. EasyUni responds by sorting all list items according to their salary and displaying it to the user. |
| *Post-Condition:*   * All courses in the system have been successfully displayed and are sorted according to their salary. |
| *Alternate flows:*  AF-S1: Sorted by Descending Salary   1. List items will be sorted from descending order according to their Salary |
| *Exceptions:*  Ex 1: GOV.SG API fails to respond   1. EasyUni will display “Unable to Sort by Salary, try again later!” 2. EasyUni will not sort list items according to salary, but will display Courses in alphabetical order. |

|  |
| --- |
| *Use Case Name:* View Course |
| *Participating Actors:* Guest, GOV.SG API, Google Map |
| *Pre-Condition:*   * Guest must have selected the course they wish to view |
| *Flow of events:*   1. Guest select the course they wish to view on the website. 2. EasyUni request salary data from GOV.SG API. 3. GOV.SG API responds by providing salary data for the course. 4. EasyUni request map data from Google Map. 5. Google Map responds by providing map data. 6. EasyUni responds by presenting the course details to the user. *The page contains all course details such as; course name, IGP, salary, school name, course rating & number of people rated, IGP time series graph, salary time series graph and google map details.* |
| *Post-Condition:*   * All course details have been successfully displayed. |
| *Alternate flows:* |
| *Exceptions:*  Ex 1: GOV.SG API fails to respond   1. EasyUni will display “Salary not found” on the salary data column instead.   Ex 2: Google Map fails to respond   1. EasyUni will display “Location not found” on the location column instead. |

|  |
| --- |
| *Use Case Name:* View Discussions |
| *Participating Actors:* Guest |
| *Pre-Condition:* |
| *Flow of events:*   1. Guest clicks ‘View Discussions’ on the website. 2. EasyUni responds by presenting the list of discussions to the user. *The list items include; discussion name, creator name, date, number of views and number of replies.* |
| *Post-Condition:*   * All discussions in the system have been successfully displayed. |
| *Alternate flows:* |
| *Exceptions:*  Ex 1: There are no discussions in the list   1. EasyUni will display “There are no discussions, would you like to create one?” with a ‘new post’ button instead. |

|  |
| --- |
| *Use Case Name:* Edit Profile |
| *Participating Actors:* Registered User |
| *Pre-Condition:*   * Registered User is on their profile page |
| *Flow of Events:*   1. Registered User clicks ‘edit profile’ on the website 2. EasyUni responds by presenting a profile form to the user. *The form is pre-filled in with the user’s details that were set before.* 3. Registered User updates the details in the form 4. Registered User clicks on ‘save profile’ on the form |
| *Post-Condition:*   * Registered User has successfully updated their profile * Registered User chooses to cancel and undo any edits made |
| *Alternate flows:*  AF-S1: User attempts to leave the page without saving   1. EasyUni will prompt the user “Are you sure? Any changes made will not be saved.” to the user. 2. Return to step 2. |
| *Exceptions:*  Ex 1: System failed to capture the details into database   1. EasyUni will prompt user the error “something went wrong. Please try again later!” 2. Return to Step 2. |

|  |
| --- |
| *Use Case Name:* New Discussion Topic |
| *Participating Actors:* Registered User |
| *Pre-Condition:*   * Registered User is in the Discussion page |
| *Flow of Events:*   1. Registered User clicks on ‘new post’ on the website 2. EasyUni responds by presenting a textbox and a title prompt for the user to fill in. *Apart from those features, the Topic title, Registered User name and Current date is displayed as well.* 3. Registered User enters their discussion content into the textbox and fills in the title 4. Registered User clicks on ‘Submit’ |
| *Post-Condition:*   * Registered User has successfully created a new Discussion post * Registered User cancels the creation of the new Discussion post |
| *Alternate flows:*  AF-S1: User attempts to leave the page without saving   1. EasyUni will prompt the user “Are you sure? Any changes made will not be saved.” to the user. |
| *Exceptions:*  Ex 1: System failed to capture the details into database   1. EasyUni will prompt user the error “Something went wrong. Please try again later!” 2. Return to step 2 |

|  |
| --- |
| *Use Case Name:* Review Course |
| *Participating Actors:* Registered User |
| *Pre-Condition:*   * Registered User is in the Course page |
| *Flow of Events:*   1. Registered User clicks on ‘post review’ on the website 2. EasyUni responds by presenting a textbox for the user to fill in 3. Registered User enters their review of the course into the textbox 4. Registered User clicks on ‘Submit’ |
| *Post-Condition:*   * Registered User has successfully created a new Course review |
| *Alternate flows:*  AF-S1: User attempts to leave the page without saving   1. EasyUni will prompt the user “Are you sure? Any changes made will not be saved.” to the user.   AF-S3: User cancels the creation of the new Course review   1. User clicks on ‘Cancel’ 2. EasyUni redirects back to the Course page that the User was previously on |
| *Exceptions:*  Ex 1: System failed to capture the details into database   1. EasyUni will prompt user the error “something went wrong. Please try again later!” 2. Return to Step 2. |

|  |
| --- |
| *Use Case Name:* Report Discussion |
| *Participating Actors:* Registered User |
| *Pre-Condition:*   * Registered User is viewing a Discussion topic posted in the Discussion page |
| *Flow of Events:*   1. Registered User clicks on ‘Report’ on the original post or following comments 2. EasyUni responds by presenting a report form to the Registered User. *The form contains multiple options for the Registered User to select the reason for reporting: use of slurs, inappropriate remarks, trolling, disinformation, and a others: option for the Registered User to fill in if none of them apply* |
| *Post-Condition:*   * Registered User successfully completes the report form |
| *Alternate flows:*  AF-S1: User attempts to leave the page without saving   1. EasyUni will prompt the user “Are you sure? Any changes made will not be saved.” to the user.   AF-S2: User cancels the filling of the report form   1. User clicks on ‘Cancel’ 2. EasyUni redirects back to the Discussion topic page that the User was previously on |
| *Exceptions:*  Ex 1: System failed to capture the details into database   1. EasyUni will prompt user the error “Something went wrong. Please try again later!” 2. Return to Step 2. |

|  |
| --- |
| *Use Case Name:* Reply to Discussion |
| *Participating Actors:* Registered User |
| *Pre-Condition:*   * Registered User is viewing a Discussion topic posted in the Discussion page |
| *Flow of Events:*   1. Registered User clicks on ‘Reply’ on the website 2. EasyUni responds by presenting a textbox for the user to fill in. *Apart from the textbox, the Topic title, Registered User name and Current date is displayed as well.* 3. Registered User fills in the textbox with their reply to the topic post 4. Registered User clicks on ‘Submit’ |
| *Post-Condition:*   * Registered User has successfully created a new reply * Registered User cancels the creation of the new reply |
| *Alternate flows:*  AF-S1: User attempts to leave the page without saving   1. EasyUni will prompt the user “Are you sure? Any changes made will not be saved.” to the user. |
| *Exceptions:*  Ex 1: System failed to save response from the Registered Users   1. EasyUni will display the message “Failed to submit,please try again” in 2 seconds. |

|  |
| --- |
| *Use Case Name:* Delete Reply |
| *Participating Actors:* Admin |
| *Pre-Condition:*   * Admin is viewing a Discussion topic posted in the Discussion page |
| *Flow of Events:*   1. Admin clicks ‘Delete’ on a particular reply box 2. EasyUni responds by presenting a delete form to the Admin. *The delete form will have a textbox for the Admin to specify the reason for deletion* 3. Admin completes the delete form by filling out the reason 4. Admin clicks ‘Submit’ |
| *Post-Condition:*   * Reply has been successfully deleted from the discussion post * Admin cancels the deletion |
| *Alternate flows:*  AF-S3: Admin cancels the deletion   1. Admin clicks on ‘Cancel’ 2. EasyUni redirects back to the Discussion topic page that the Admin was previously on |
| *Exceptions:*  EX1: Reply has already been deleted   1. EasyUni displays an error message that says “Error, this discussion post no longer exists” for 3 seconds 2. EasyUni redirects back to the Discussion thread page that the Admin was previously on |

|  |
| --- |
| *Use Case Name:* Delete Discussion |
| *Participating Actors:* Admin |
| *Pre-Condition:*   * Admin is viewing the Discussion page |
| *Flow of Events:*   1. Admin clicks ‘Delete’ on a particular discussion thread 2. EasyUni responds by presenting a delete form to the Admin. *The delete form will have a textbox for the Admin to specify the reason for deletion* 3. Admin completes the delete form by filling out the reason 4. Admin clicks ‘Submit’ |
| *Post-Condition:*   * Discussion thread has been successfully deleted from the discussion page * Admin cancels the deletion |
| *Alternate flows:*  AF-S3: Admin cancels the deletion   1. Admin clicks on ‘Cancel’ 2. EasyUni redirects back to the Discussion page that the Admin was previously on |
| *Exceptions:*  EX1: Discussion thread has already been deleted   1. EasyUni displays an error message that says “Error, this discussion thread no longer exists” for 3 seconds 2. EasyUni redirects back to the Discussion page that the Admin was previously on |

|  |
| --- |
| *Use Case Name:* Manage User |
| *Participating Actors:* Admin |
| *Pre-Condition:*   * *Must be logged in as admin* |
| *Flow of Events:*   1. EasyUni shows all the available Users. 2. Admin select any user profiles to view. |
| *Post-Condition:*   * Admin selected a user profile to view |

|  |
| --- |
| *Use Case Name:* Delete User |
| *Participating Actors:* Admin |
| *Pre-Condition:*   * Admin is viewing a user profile page |
| *Flow of Events:*   1. Admin clicks ‘Delete’ 2. EasyUni responds by presenting a delete form to the Admin. *The delete form will have a textbox for the Admin to specify the reason for deletion* 3. Admin completes the delete form by filling out the reason 4. Admin clicks ‘Submit’ |
| *Post-Condition:*   * User has been successfully deleted from the system |
| *Alternate flows:*  AF-S3: Admin cancels the deletion   1. Admin clicks on ‘Cancel’ 2. EasyUni redirects back to the user profile page that the Admin was previously on |
| *Exceptions:*  EX1: >1 Admin user tries to delete the same registered user at roughly the same time   1. EasyUni displays an error message that says “Error, this user no longer exists” for 3 seconds 2. EasyUni redirects back to the homepage |

|  |
| --- |
| *Use Case Name:* Delete Course Review |
| *Participating Actors:* Admin |
| *Pre-Condition:*   * Admin is viewing a course review on a course detail page |
| *Flow of Events:*   1. Admin clicks ‘Delete’ 2. EasyUni responds by presenting a delete form to the Admin. *The delete form will have a textbox for the Admin to specify the reason for deletion* 3. Admin completes the delete form by filling out the reason and additional details if necessary 4. Admin clicks ‘Submit’ |
| *Post-Condition:*   * Course review has been successfully deleted from the system |
| *Alternate flows:*  AF-S3: Admin cancels the deletion   1. Admin clicks on ‘Cancel’ 2. EasyUni redirects back to the course review page that the Admin was previously on |
| *Exceptions:*  EX1: Review has already been deleted   1. EasyUni displays an error message that says “Error, this course review no longer exists” for 3 seconds 2. EasyUni redirects back to the course review page that the Admin was previously on |

|  |
| --- |
| *Use Case Name:* Send Verification Email |
| *Participating Actors:* Email Server |
| *Pre-Condition:*   * This use case extends the Register use case, which is initiated by the Guest registering for a new account * Email provided by the guest is verified by EasyUni to have no existing account tied to it |
| *Flow of Events:*   1. Email Server sends an email to the one specified by the guest via the register form |
| *Post-Condition:*   * Verification email has been successfully sent out |
| *Alternate flows:* |
| *Exceptions:*  EX1: Guest email is unable to receive verification email (for several possible reasons - invalid email address, email is full etc.)   1. EmailServer sends an error log to EasyUni indicating that it was unable to send the email 2. EasyUni displays the message “Unable to send verification email to <insert\_guest\_email\_here>, please check if it is a valid and working email account.” for 5 seconds |

|  |
| --- |
| *Use Case Name:* Send Reset Password Email |
| *Participating Actors:* Email Server |
| *Pre-Condition:*   * This use case extends the Reset Password use case, which is initiated by the Guest upon clicking ‘Reset Password’ * Guest has filled out the reset password form with their email and that has been verified by EasyUni |
| *Flow of Events:*   1. Email Server sends out an email with detailed instructions on how to reset their password to the email address provided. |
| *Post-Condition:*   * Reset password email has been successfully sent out |
| *Alternate flows:* |
| *Exceptions:*  EX1: Guest email is unable to receive reset password email (for several possible reasons - invalid email address, email is full etc.)   1. EmailServer sends an error log to EasyUni indicating that it was unable to send the email 2. EasyUni displays the message “Unable to send reset password email to <insert\_guest\_email\_here>, please check if it is a valid and working email account.” for 5 seconds |

|  |
| --- |
| *Use Case Name:* Display School on Map |
| *Participating Actors:* Google Map |
| *Pre-Condition:*   * This use case extends the View Course use case, which is initiated by the Guest accessing the course detail page |
| *Flow of Events:*   1. EasyUni requests map data from Google Map. 2. Google Map responds by providing map data. |
| *Post-Condition:*   * Google Map successfully returns the map data * Google Map is unable to return the map data |
| *Alternate flows:* |
| *Exceptions:*  EX1: Google Map is unable to properly return the map data   1. EasyUni detects that no data or corrupted data has been received. 2. EasyUni displays an error message that says “Unable to display map” for 3 seconds. 3. EasyUni will display “Location not found” on the location column instead. |

|  |
| --- |
| *Use Case Name:* Provide Salary Data |
| *Participating Actors:* GOV.SG API |
| *Pre-Condition:*   * This use case extends the View Courses and View Course use cases. * Initiated whenever the Guest wishes to access and view salary data |
| *Flow of Events:*   1. EasyUni request salary data from GOV.SG API. 2. GOV.SG API responds by providing salary data for the course. |
| *Post-Condition:*   * GOV.SG API successfully returns the salary data |
| *Alternate flows:* |
| *Exceptions:*  EX1: GOV.SG API is unable to properly return the salary data   1. EasyUni detects that no data or corrupted data has been received. 2. EasyUni displays an error message that says “Unable to receive salary data” for 3 seconds. 3. EasyUni will display “Salary not found” on the salary data column instead. |