project proposal

Momento - Journal application for sentimental analysis

Nanyang Technological University

# Content page

[Content page 1](#_Toc461564226)

[Background 4](#_Toc461564227)

[Causes of depression 4](#_Toc461564228)

[Consequences of depression 4](#_Toc461564229)

[Ways to prevent or treat depression 4](#_Toc461564230)

[Prevalence of depression 4](#_Toc461564231)

[Problem statement 5](#_Toc461564232)

[Our solution 5](#_Toc461564233)

[Application details 5](#_Toc461564234)

[Implementation 6](#_Toc461564235)

[Project timeline 6](#_Toc461564236)

[Requirement Elicitation 6](#_Toc461564237)

[Requirement Analysis 7](#_Toc461564238)

[Design & Implementation 8](#_Toc461564239)

[Technical specification 9](#_Toc461564240)

[Mission statement 9](#_Toc461564241)

[Functional requirement 9](#_Toc461564242)

[School administration features 9](#_Toc461564243)

[Student features 9](#_Toc461564244)

[Teacher features 9](#_Toc461564245)

[Counselors features 10](#_Toc461564246)

[Non-functional requirement 10](#_Toc461564247)

[Usability 10](#_Toc461564248)

[Reliability 10](#_Toc461564249)

[Performance 10](#_Toc461564250)

[Supportability 11](#_Toc461564251)

[Security 11](#_Toc461564252)

[Data dictionary 11](#_Toc461564253)

[Account type 11](#_Toc461564254)

[Admin account 11](#_Toc461564255)

[Application session 11](#_Toc461564256)

[Assignment 11](#_Toc461564257)

[Comment 11](#_Toc461564258)

[Counsellor account 11](#_Toc461564259)

[Depression classification 12](#_Toc461564260)

[False-positive 12](#_Toc461564261)

[Journal entry 12](#_Toc461564262)

[Module 12](#_Toc461564263)

[Student account 12](#_Toc461564264)

[Teacher account 12](#_Toc461564265)

[User session 12](#_Toc461564266)

[Wait time 12](#_Toc461564267)

[Use case diagrams 13](#_Toc461564268)

[Common features 13](#_Toc461564269)

[Administration features 13](#_Toc461564270)

[Student features 14](#_Toc461564271)

[Teacher features 14](#_Toc461564272)

[Counsellor/Therapist features 15](#_Toc461564273)

[Use case description 16](#_Toc461564274)

[Common features 16](#_Toc461564275)

[Login 16](#_Toc461564276)

[Logout 17](#_Toc461564277)

[View a student account 18](#_Toc461564278)

[View detailed metrics of feature sets 19](#_Toc461564279)

[Indicate false positive identification of student account 20](#_Toc461564280)

[Administration features 21](#_Toc461564281)

[Assign modules to students 21](#_Toc461564282)

[Unassign modules from students 22](#_Toc461564283)

[View teacher account 23](#_Toc461564284)

[Assign students 24](#_Toc461564285)

[Unassign students 25](#_Toc461564286)

[Student features 26](#_Toc461564287)

[Create journal entry 26](#_Toc461564288)

[View journal entry (Student) 27](#_Toc461564289)

[Reply to comments 28](#_Toc461564290)

[Link to Facebook 29](#_Toc461564291)

[Link to Twitter 30](#_Toc461564292)

[Teacher features 31](#_Toc461564293)

[View list of assigned student accounts 31](#_Toc461564294)

[Filter students 32](#_Toc461564295)

[View journal entry (Teacher) 33](#_Toc461564296)

[Enter comments 34](#_Toc461564297)

[Search nearby counsellors 35](#_Toc461564298)

[View counsellor account 37](#_Toc461564299)

[Share student accounts 38](#_Toc461564300)

[Set grades for assigned students 39](#_Toc461564301)

[Counsellor/Therapist features 40](#_Toc461564302)

[View list of shared student account 40](#_Toc461564303)

[View treatment logs 41](#_Toc461564304)

[Enter treatment log 42](#_Toc461564305)

[View journal entry (Counsellor) 43](#_Toc461564306)

[Conclusion 44](#_Toc461564307)

[References 45](#_Toc461564308)

# Background

Depression is a mood disorder that affects how people think, behave and feel in a negative manner.

## Causes of depression

There are many factors that can contribute to depression. Some of the possible factors are family problems, stress and even negative thinking patterns.

## Consequences of depression

Depression usually causes the victim to feel lethargic constantly which reduces the victim’s efficiency in both work and life.

It also increases irritability, anger and anxiety of the victim, hindering their social lifestyle.

It has also been clinically proven that depression can lead to several other medical conditions, such as cancer, infections, heart conditions and even neurological diseases such as brain tumors or Parkinson’s disease in a long run.

The worst case scenario for depression, however, is the very fact that depression sufferers usually entertain delusions and false beliefs that they are worthless and may lead to fatal incidents such as suicide.

## Ways to prevent or treat depression

There are multiple ways to treat depression. Depression can be treated via short term methods such as medication or it can also be treated via long term methods such a counselling, therapies and even changes in lifestyle. These treatment methods coupled with social support usually result in high recovery rate.

## Prevalence of depression

It is estimated that 6.3% of the population in Singapore are affected by depression during their lifetime [1]. A large percentage of this figure falls into the children and adolescent age category.

# Problem statement

From the background information, we know the seriousness of depression, its consequences and ways to treat it.

The main problem is that most depression sufferers find themselves unable to relate their condition to people around them and as a result will choose to keep it to themselves. This causes a delay in the detection and treatment of the depression.

In some cases, the depression sufferers themselves might not even know that they are suffering from the condition. This can potentially result in their depression going on untreated, leading to undesired outcomes that can be as worse as suicide attempts.

# Our solution

Momento is a web application that focuses on solving the stated problem for people in the children or adolescent age category*.* The main idea of the solution leverages on the fact that most schools include in their curriculum the requirement for their students to write journals on a periodic basis. Coupling the journals written by students and the recent advancements of machine learning for the specific focus of sentimental analysis, we are able to obtain a depression index. On top of the feature of journal sentimental analysis, we are also able to include other feature sets such as the student’s grade delta and also depression index from feeds obtained through social media accounts of the student. With all these metrics, we will be able to highlight to the student’s form teacher whenever we detect any anomaly in the student’s overall depression status, so that appropriate action can be undertaken to aid the student in recovery.

## Application details

The main users of the application will be the students, teachers and counselors/therapists. The application will show different information for different types of users.

For students, the application will be a simplistic journal application where they will be able to write down their thoughts and experiences for their daily lives on a periodic basis.

For teachers, the application will feature a simple indicator to highlight students that requires extra attention. The application will also categorize student into 5 main categories, Strongly Depressed, Depressed, Neutral, Motivated and Highly Motivated. This form of categorization aids the teachers in deciding time allocation for different students based on their individual needs.

For counselors/therapists, the application will feature a detailed overview of the student’s mental health along with the gathered statistics from all the different feature sets. This guides the counselors/therapists in formulating ways to help the student recover from their depression in the shortest time possible.

# Implementation

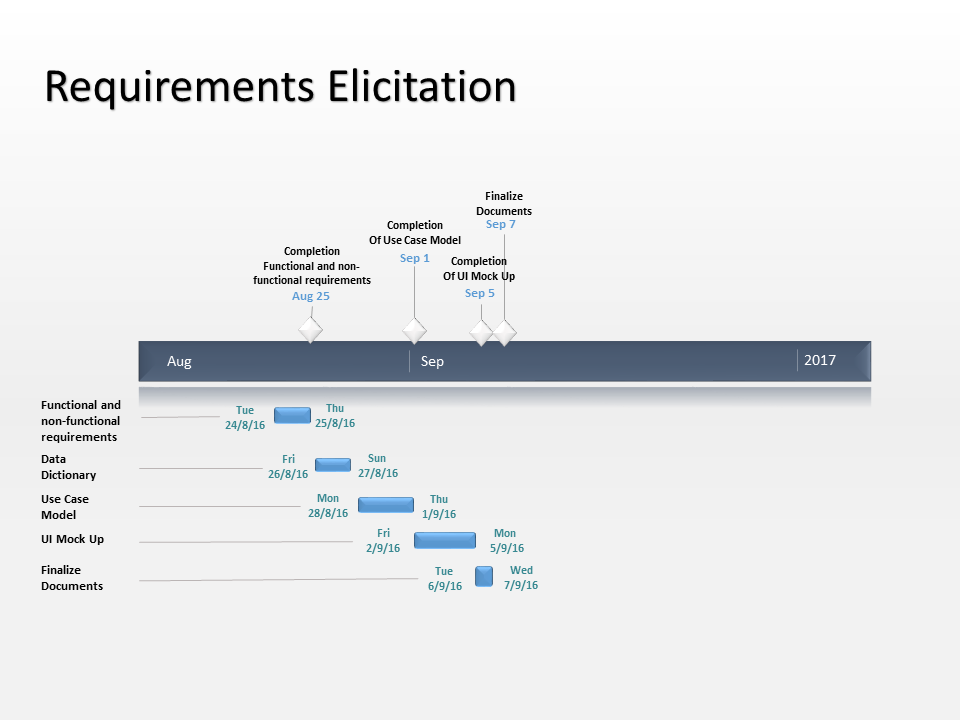
The implementation of Momento will be split into 3 main phases:

* Requirement Elicitation
* Requirement Analysis
* Design & Implementation

## Project timeline

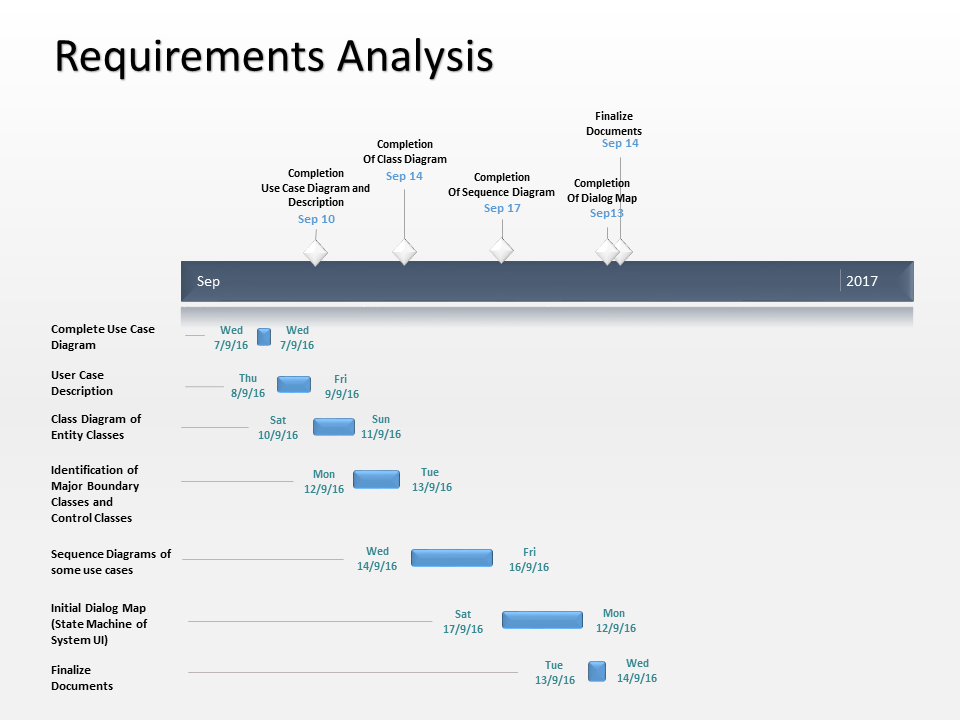
### Requirement Elicitation

This phase focuses on the elicitation of the required functionality and features for Momento.



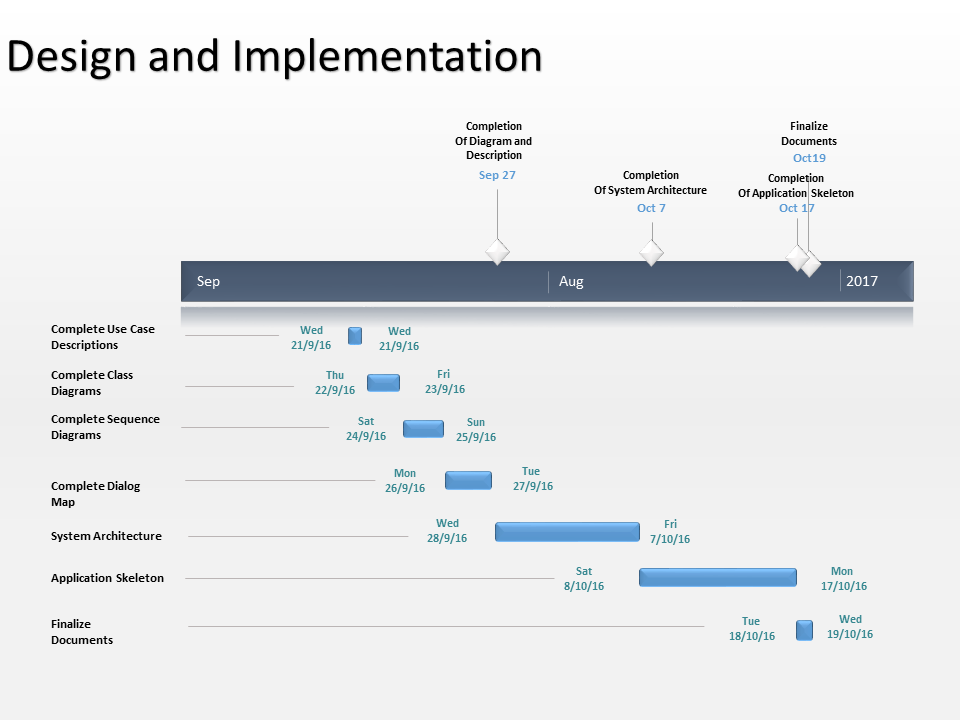
### Requirement Analysis

This phase focuses on the analysis of the features decided in the requirement elicitation phase and designing the user interface mockups as well as system architecture.



### Design & Implementation

This phase focuses on the implementation of the designed system architecture and building of the feature decided in the requirement elicitation phase.



# Technical specification

## Mission statement

Team 6 will develop a web application and an android application that is capable of early detection of depression. The detection of depression will be done by analyzing metrics of the different feature sets using machine learning technique. The feature sets consist of the journal entries, grades and social media feeds of the students. The application will be used by 4 types of users, school administrators, students, teachers and counsellors/therapists. Different types of users will be able to access different functional features catered to suit their respective needs. This application can increase the rate of early detection for depression and improve overall recovery rate for the depression sufferers.

## Functional requirement

### School administration features

* The school administrator must be able to login to their admin account
* The school administrator must be able to logout from their admin account
* The school administrator must be able to assign multiple student account to a teacher account
* The school administrator must be able to unassign multiple student account from a teacher account
* The school administrator must be able to assign multiple students to multiple modules

### Student features

* The student must be able to login to their student account
* The student must be able to logout from their student account
* The student must be able to enter their journal entry
* The student must be able to link their Facebook account to their student account
* The student must be able to link their Twitter account to their student account
* The student must be able to view their past journal entries according to the date posted

### Teacher features

* The teacher must be able to login to their teacher account
* The teacher must be able to logout from their teacher account
* The teacher must be able to view the list of student account assigned to them
* The teacher must be able to set the grades of the student account assigned to them for all modules the student is assigned
* The teacher must be able to view the journal entries in the student account assigned to them
* The teacher must be able to enter comments for a journal entry in the student account assigned to them
* The teacher must be able to view detailed information of a particular student account assigned to them
* The teacher must be able to view an overview of all student accounts assigned to them and their respective depression classification that the student account is under
* The teacher must be able to filter student accounts based on their depression classification
* The teacher must be able to view detailed metrics of the feature sets that contributes to the depression classification of a particular student account assigned to them
* The teacher must be able to share student account information with a counselor account
* The teacher must be able to search nearby counselors or therapists
* The teacher must be notified when any student account assigned under them requires immediate attention as determined by the system’s depression analytics machine
* The teacher must be able to indicate false-positive identification of depression classification of a particular student account to the system

### Counselors features

* The counselor must be able to login to their counselor account
* The counselor must be able to logout from their counselor account
* The counselor must be able to view a list of all students account that is shared with them
* The counselor must be able to view detailed metrics of the feature sets that contributes to the depression classification of a particular student account shared with them
* The counselor must be able to view the journal entries in the student account shared with them
* The counselor must be able to indicate false-positive identification of depression classification of a particular student account to the system
* The counselor must be able to enter treatment logs/therapy session logs into a particular student account
* The counselor must be able to view treatment logs/therapy session logs for a particular student account

## Non-functional requirement

### Usability

Student account and counsellor account must be able to retain user session across application session

Lightbox must be shown to confirm important actions made by the respective user

Lightbox must be shown to display error message when the user enters an invalid input

### Reliability

The system must be able to support concurrent access for up to 1024 users

### Performance

The system must be able to display up to date information within 5 minutes of information entry

The system must be able to perform depression status analysis for 1500 students every day

The average user wait times should not exceed 25 seconds

### Supportability

The system backend must be able to be ported across multiple google app engine VM instances without losing any functionality for the application

### Security

Teacher account and counsellor account must require the respective user to enter their username and password for every application session

## Data dictionary

### Account type

The type of an account. Can be one of the following:

1. Administrator
2. Teacher
3. Student
4. Counsellor

### Admin account

An admin account is an entity that contains a collection of records required for the administration purposes for the application it also forms a security barrier for access to specialized features of the application as described in the functional requirements - school administration features. An admin account is secured by a combination of username and password.

### Application session

An application session is defined by the time between the startup of an application and the shutdown of an application.

### Assignment

Assignment refers to the act of assigning a student to a teacher. Each student can be assigned or re-assigned at any time, but each student can only be assigned to a single teacher. The teacher whom a student is assigned to will be considered the teacher-in-charge for that particular student.

### Comment

A comment is a brief statement that can be used to provide words of encouragement or advice for the students.

### Counsellor account

A counselor/therapist account is an entity that contains a collection of records that can aid the counselor/therapist in formulating ways to help speed up the recovery progress of the student from depression. It also forms a security barrier for access to specialized features of the application as described in the functional requirements – counselor/therapist features. A counselor/therapist account is secured by a combination of username and password.

### Depression classification

A depression classification is the generalization of students into 5 main categories:

1. Strongly depressed
2. Depressed
3. Neutral
4. Motivated
5. Highly motivated

### False-positive

A condition that arises when the analytics machine wrongly classify a student into a particular depression class.

### Journal entry

A journal entry is a record that contains the words written by a student to describe their feelings or experiences through a particular time period.

### Module

A module is a self-contained course of academic study. Examples for modules in primary schools can be English, Mother Tongue and Mathematics. Examples for modules in secondary schools can be Physics, Chemistry, Biology and Geography.

### Student account

A student account is an entity that contains a collection of records such as their journal entries and personal details of a student. It also forms a security barrier for access to specialized features of the application as described in the functional requirements – student features. A student account is secured by a combination of username and password.

### Teacher account

A teacher account is an entity that contains a collection of records required to monitor the mental health of the students assigned to them. It also forms a security barrier for access to specialized features of the application as described in the functional requirements – teacher features. A teacher account is secured by a combination of username and password.

### User session

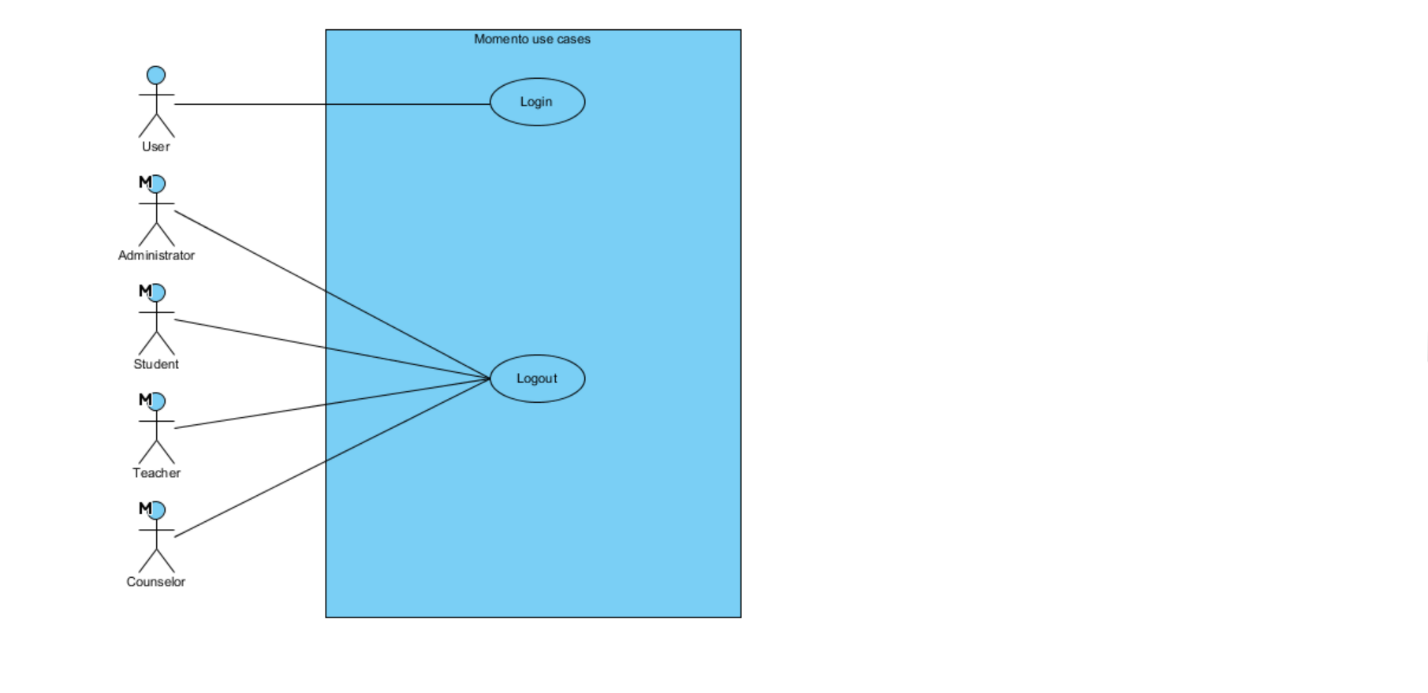
A user session is defined by the presence of a logged in user account. It is the time between a user account login and the corresponding user account logout.

### Wait time

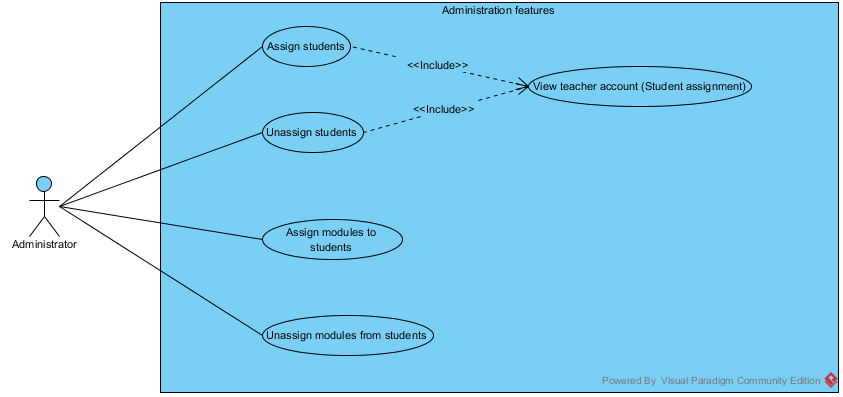
Wait time is the amount of time a user has to wait for the system’s response.

## Use case diagrams

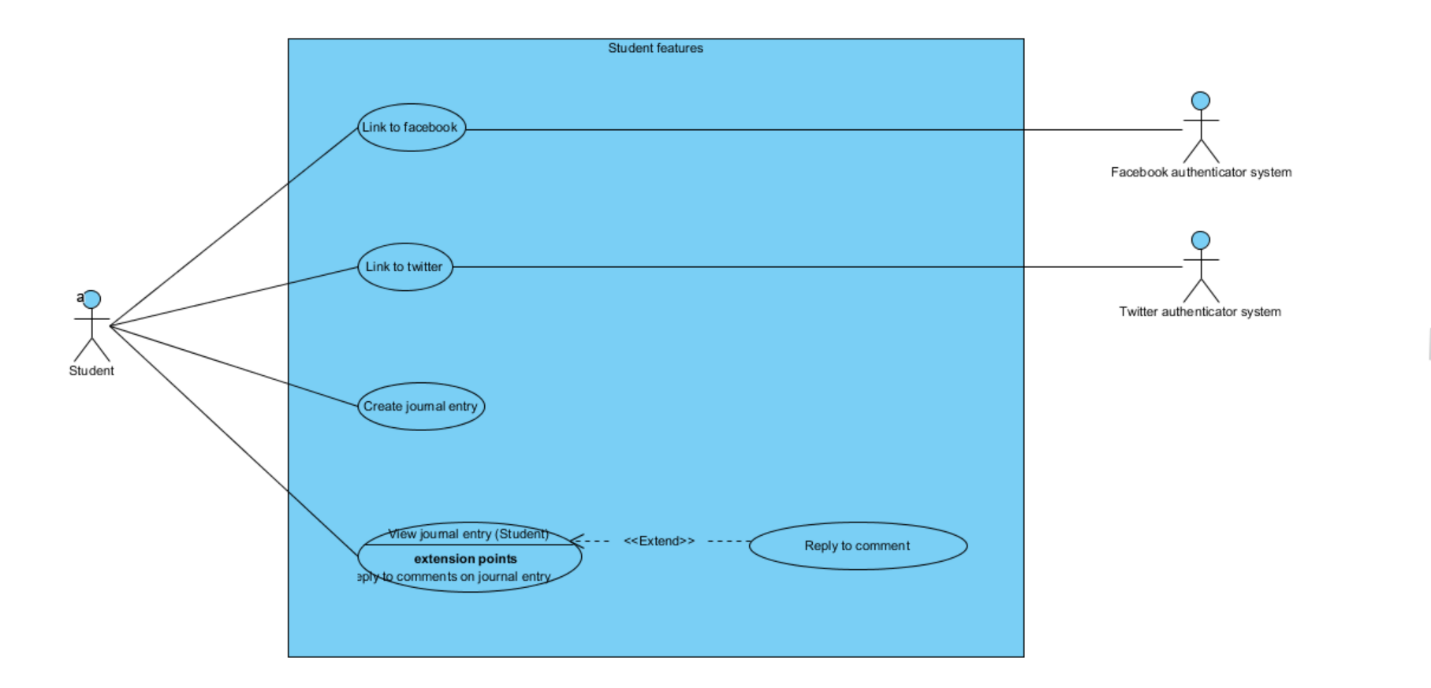
### Common features



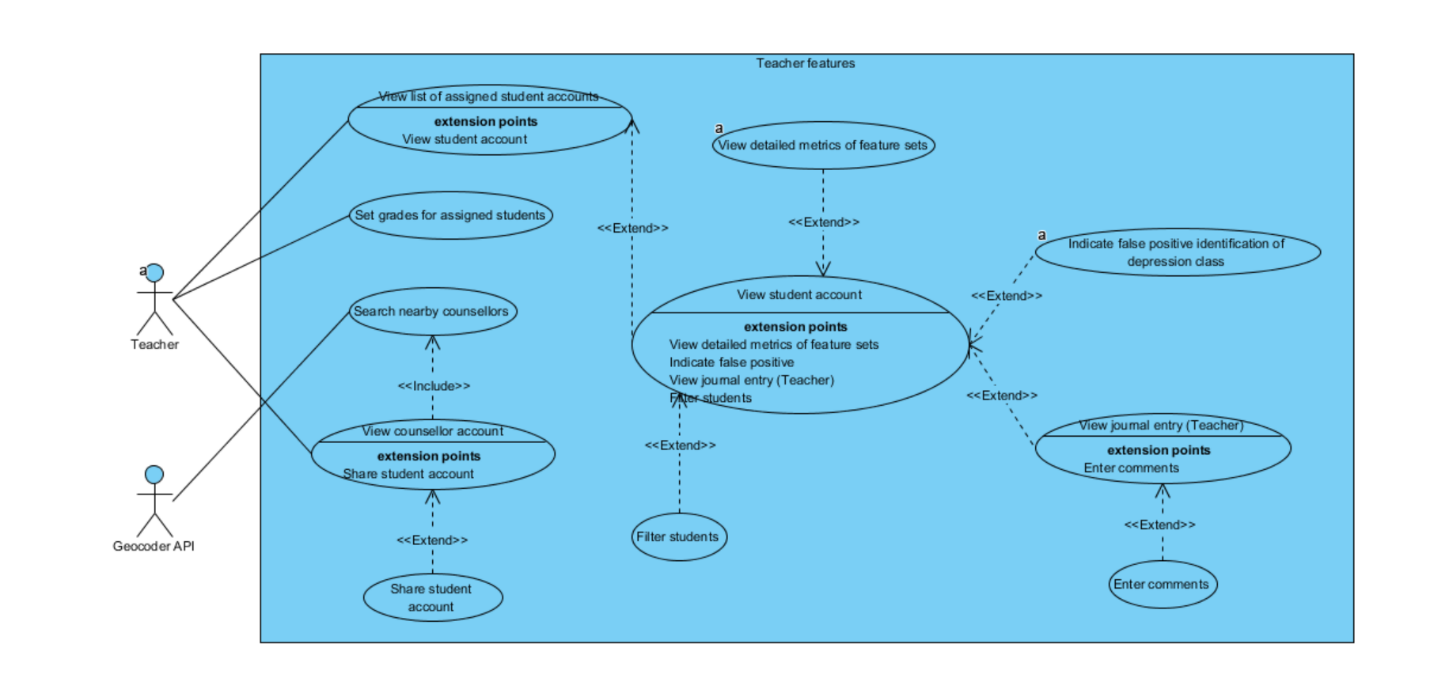
### Administration features



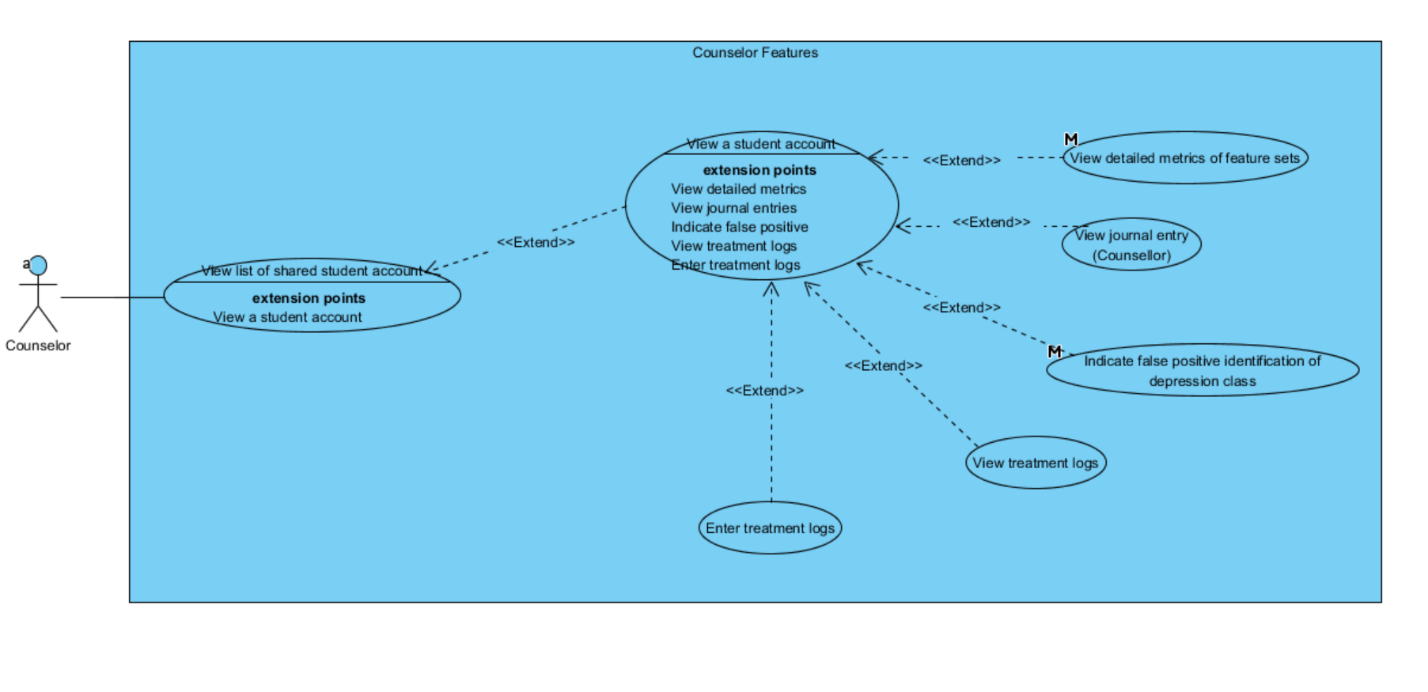
### Student features



### Teacher features



### Counsellor/Therapist features



## Use case description

### Common features

#### Login

|  |  |
| --- | --- |
| Use case no. | 1 |
| Name | Login |
| Description | User login to the system to access other sub-features specific to their account type |
| Version | 1.0.0 |
| Last edited | 05-September-2016 |
| Initiating actor(s) | User |
| Entry condition(s) | The user must have a valid account registered in the system |
| Exit condition(s) | A user session for the account type will be created in the system |
| Main flow | 1. The user launches the application 2. The system displays the login form with the following inputs:    1. Username    2. Password 3. The user completes the login form 4. The user submits the login form 5. The system verifies the username and password with the database 6. If the username, password and account type matches with the account credentials in the database the system loads the home screen for the user’s account type |
| Alternate flow | AF-S6: The username, password and account type does not match with any account credentials in the database   1. The system displays the message “Invalid username/password! Please try again!” 2. The system returns to step 2 of the main flow |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system returns to step 2 of the main flow |

#### Logout

|  |  |
| --- | --- |
| Use case no. | 2 |
| Name | Logout |
| Description | Account holder logout from the system to prevent unauthorized access to the system using their account |
| Version | 1.0.0 |
| Last edited | 05-September-2016 |
| Initiating actor(s) | Administrator, Student, Teacher, Counsellor |
| Entry condition(s) | The user must be logged in to the system |
| Exit condition(s) | The user session will be removed from the system |
| Main flow | 1. The user selects the logout option 2. The system prompts the user for confirmation of the logout action 3. The user confirms the logout action 4. The system deletes the user session for the current user from the system 5. The system informs the user logout action is successful |
| Alternate flow | AF-S3: The user rejects the logout action  The system exits the logout use case |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system exits logout use case |

#### View a student account

|  |  |
| --- | --- |
| Use case no. | 3 |
| Name | View a student account |
| Description | User views the details of a student account |
| Version | 1.0.0 |
| Last edited | 05-September-2016 |
| Initiating actor(s) | Counsellor, Teacher |
| Entry condition(s) | The user must be logged into a counsellor account or teacher account in the system  The user must have at least one student account being shared/assigned with them |
| Exit condition(s) | - |
| Main flow | 1. The user selects a student to view their account details 2. The system retrieves the student account details of the selected student 3. The system displays the student account details with the following information:    1. Student ID    2. Name    3. Age    4. Gender    5. School    6. Most recent school grades    7. Depression classification |
| Alternate flow |  |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system exits view a student account use case |

#### View detailed metrics of feature sets

|  |  |
| --- | --- |
| Use case no. | 4 |
| Name | View detailed metrics of feature sets |
| Description | User views the detailed metrics of the feature sets that contributes to the current depression classification of the selected student |
| Version | 1.0.0 |
| Last edited | 05-September-2016 |
| Initiating actor(s) | Counsellor |
| Entry condition(s) | The user must be logged in to a counsellor account or teacher account in the system  The user must have selected to view a student account |
| Exit condition(s) | - |
| Main flow | 1. The user selects view detailed metrics option 2. The system retrieves and displays the following information:    1. The trend of the selected student depression index obtained from journal entries    2. The trend of the selected student depression index obtained from Facebook feeds    3. The trend of the selected student depression index obtained from Twitter feeds    4. The trend of the selected student’s grades based on individual modules    5. The trend of the selected student’s grade normalized to the school’s grades    6. A list of most recent social media feeds and journal entries that corresponds to high depression index |
| Alternate flow | AF-S2: There are not enough data to calculate the depression metrics for the selected student   1. The system informs the user that there are not enough data present to calculate the depression metrics for the selected student |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system returns to step 2 of the main flow |

#### Indicate false positive identification of student account

|  |  |
| --- | --- |
| Use case no. | 5 |
| Name | Indicate false positive identification of student account |
| Description | User views the detailed metrics of the feature sets that contributes to the current depression classification of the selected student |
| Version | 1.0.0 |
| Last edited | 05-September-2016 |
| Initiating actor(s) | Counsellor, Teacher |
| Entry condition(s) | The user must be logged in to a counsellor or teacher account in the system  The user must have selected to view a student account |
| Exit condition(s) | The system saves the adjusted depression classification into the database  The system schedules the system to recalibrate the weights and biases of the learning algorithm based on the new depression classification |
| Main flow | 1. The user selects indicate false positive identification of student account option 2. The system displays recalibration form with the following options:    1. Strongly motivated    2. Motivated    3. Neutral    4. Depressed    5. Strongly depressed 3. The user selects the most appropriate class for the selected student and submits the form 4. The system saves the corrected depression classification into the database 5. The system schedules the system to recalibrate the weights and biases of the learning algorithm based on the new depression classification 6. The system informs the user that the depression classification has been changed |
| Alternate flow | - |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system returns to step 2 of the main flow |

### Administration features

#### Assign modules to students

|  |  |
| --- | --- |
| Use case no. | 6 |
| Name | Assign modules to students |
| Description | Administrator assigns multiple modules to multiple students |
| Version | 1.0.0 |
| Last edited | 05-September-2016 |
| Initiating actor(s) | Administrator |
| Entry condition(s) | The user must be logged in to an administrator account in the system |
| Exit condition(s) | The assignment records of modules to students will be saved into the database |
| Main flow | 1. The user selects assign modules to students use case 2. The system retrieves and displays an assignment form that contains a list of students and a list of modules    1. Each student in the student list contains the following details:       1. Student ID       2. Student name    2. Each module in the module list contains the following details:       1. Module code       2. Module name 3. The user selects multiple students from the student list 4. The user selects multiple modules from the module list to assign to the selected students 5. The user submits the assignment form 6. The system creates an assignment record for each combination of module and student selected 7. The system saves the assignment record in the database 8. The system informs the user that the operation is a success |
| Alternate flow | - |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system exits logout use case |

#### Unassign modules from students

|  |  |
| --- | --- |
| Use case no. | 7 |
| Name | Unassign modules from students |
| Description | Administrator unassign multiple modules from multiple students |
| Version | 1.0.0 |
| Last edited | 05-September-2016 |
| Initiating actor(s) | Administrator |
| Entry condition(s) | The user must be logged in to an administrator account in the system |
| Exit condition(s) | The assignment records of modules to students will be saved into the database |
| Main flow | 1. The user selects unassign modules from students use case 2. The system retrieves and displays a list of all students with each student containing the following information:    1. Student ID    2. Student name 3. The user selects multiple students to unassign modules 4. The system retrieves and displays a list of all modules that are assigned to all the selected students with each module containing the following information:    1. Module code    2. Module name 5. The user selects the modules to be unassigned from the selected students and submits 6. The system displays a summary of the action and prompts the user for confirmation 7. The user confirms the action 8. The system deletes the assignment records for each combination of student and modules selected 9. The system informs user that the operation is a success |
| Alternate flow | AF-S6: The user rejects the action   1. The system returns to step 2 of the main flow |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system exits logout use case |

#### View teacher account (Student assignment)

|  |  |
| --- | --- |
| Use case no. | 8 |
| Name | View teacher account (Student assignment) |
| Description | Administrator views a teacher account |
| Version | 1.0.0 |
| Last edited | 05-September-2016 |
| Initiating actor(s) | Administrator account |
| Entry condition(s) | The user must be logged in to an administrator account in the system |
| Exit condition(s) | - |
| Main flow | 1. The user selects view teacher account option 2. The system retrieves and displays a list of teachers that belong to the same school as the administrator account. The list of teacher accounts contains the following information:    1. Teacher ID    2. Teacher name 3. The user selects a teacher to view teacher details 4. The system retrieves and displays the teacher account with the following details:    1. Teacher ID    2. Teacher name    3. School    4. Age    5. Gender |
| Alternate flow | - |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system exits view a student account use case |

#### Assign students

|  |  |
| --- | --- |
| Use case no. | 9 |
| Name | Assign students |
| Description | Administrator assigns multiple students to a teacher account |
| Version | 1.0.0 |
| Last edited | 05-September-2016 |
| Initiating actor(s) | Administrator account |
| Entry condition(s) | The user must have selected to view a teacher account |
| Exit condition(s) | The assignment records of students to the selected teacher will be saved into the database |
| Main flow | 1. The user selects assign students option 2. The system retrieves and displays a list of unassigned students that belong to the same school as the administrator account. The list of student accounts contains the following information:    1. Student ID    2. Name    3. Age    4. Gender 3. The user selects multiple students to assign to the teacher 4. The user submits the assignment 5. The system creates an assignment record for the selected teacher and each student selected 6. The system saves the assignment record in the database 7. The system informs the user that the operation is a success |
| Alternate flow | - |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system exits view a student account use case |

#### Unassign students

|  |  |
| --- | --- |
| Use case no. | 10 |
| Name | Unassign students |
| Description | Administrator unassigns multiple students from a teacher account |
| Version | 1.0.0 |
| Last edited | 05-September-2016 |
| Initiating actor(s) | Administrator account |
| Entry condition(s) | The user must have selected to view a teacher account |
| Exit condition(s) | The assignment records of students to the selected teacher will be deleted from the database |
| Main flow | 1. The user selects assign students option 2. The system retrieves and displays a list of students assigned to the selected teacher account. The list of student accounts contains the following information:    1. Student ID    2. Name    3. Age    4. Gender 3. The user selects multiple students to unassign from the teacher and submits 4. The system displays a summary of the action and prompts the user for confirmation 5. The user confirms the action 6. The system deletes the assignment records for the selected students and the selected teacher 7. The system informs user that the operation is a success |
| Alternate flow | AF-S6: The user rejects the action   1. The system returns to step 2 of the main flow |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system exits view a student account use case |

### Student features

#### Create journal entry

|  |  |
| --- | --- |
| Use case no. | 11 |
| Name | Create journal entry |
| Description | User creates a new journal entry |
| Version | 1.0.0 |
| Last edited | 05-September-2016 |
| Initiating actor(s) | Student |
| Entry condition(s) | The user must be logged in to a student account in the system |
| Exit condition(s) | The system creates and saves a new journal entry into the database |
| Main flow | 1. The user selects create journal entry option 2. The system displays a journal entry form with the following fields:    1. Title    2. Journal 3. The user completes the journal entry form 4. The user submits the journal entry form 5. The system verifies that all fields are completed 6. The system creates a new journal entry with the details entered in the journal entry form, along with the current date and time 7. The system saves the journal entry into the database 8. The system informs the user that the journal entry is saved |
| Alternate flow | AF-S5: Some fields in the journal entry form are not completed   1. The system prompts the user to complete the journal entry form before submitting 2. The system returns to step 2 of the main flow |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system returns to step 2 of the main flow |

#### View journal entry (Student)

|  |  |
| --- | --- |
| Use case no. | 12 |
| Name | View journal entry (Student) |
| Description | Student views past journal entries submitted by themselves |
| Version | 1.0.0 |
| Last edited | 05-September-2016 |
| Initiating actor(s) | Student |
| Entry condition(s) | The user must be logged in to a student account in the system |
| Exit condition(s) | - |
| Main flow | 1. The user selects view journal entries option 2. The system retrieves a list of journal entries submitted by the user 3. The system displays the list of journal entries in chronological order, with each journal entry having the following detail:    1. Date    2. Time    3. Title 4. The user selects a journal entry 5. The system retrieves details of the selected journal entry 6. The system displays the journal entry with the following detail:    1. Date    2. Time    3. Journal    4. Comments made by teacher    5. Reply made by student |
| Alternate flow | - |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system exits view journal entry use case |

#### Reply to comments

|  |  |
| --- | --- |
| Use case no. | 13 |
| Name | Reply to comment |
| Description | Student reply to comments made by teacher for a particular journal entry |
| Version | 1.0.0 |
| Last edited | 05-September-2016 |
| Initiating actor(s) | Student |
| Entry condition(s) | The student must have selected to view a journal entry  The selected journal entry must have a comment |
| Exit condition(s) | The system saves the reply into the database |
| Main flow | 1. The user selects reply to comment option 2. The system displays a reply form with the following field:    1. Reply 3. The user completes the form 4. The user submits the form 5. The system verifies that all fields are completed 6. The system creates a new reply with the details entered in the reply form, along with details such as:    1. Date    2. Time    3. Comment ID of the comment that the student is replying to 7. The system saves the reply into the database 8. The system informs the user that the reply is saved |
| Alternate flow | AF-S5: Some fields in the reply form are not completed   1. The system prompts the user to complete the reply form before submitting 2. The system returns to step 2 of the main flow |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system exits logout use case |

#### Link to Facebook

|  |  |  |  |
| --- | --- | --- | --- |
| Use case no. | 14 | | |
| Name | Link to Facebook | | |
| Description | Student links their Facebook account to the system | | |
| Version | 1.0.0 | | |
| Last edited | 05-September-2016 | | |
| Initiating actor(s) | Student | Supporting Actor(s) | Facebook authentication system |
| Entry condition(s) | The user must be logged in to a student account in the system | | |
| Exit condition(s) | The student account is updated with the OAuth token for their Facebook account in the database | | |
| Main flow | 1. The user selects link to Facebook option 2. The system passes control over to Facebook authentication system 3. The user interacts with the Facebook authentication system to complete authentication process 4. The Facebook authentication system passes control back to the system with an OAuth token 5. The system updates the user’s student account with the retrieved OAuth token in the database 6. The system informs the user that their Facebook account has been linked to their student account | | |
| Alternate flow | AF-S4: Facebook authentication system returns an error to the system   1. The system informs the user the error that has occurred during the authentication process. 2. The system exits link to Facebook use case | | |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system exits logout use case | | |

#### Link to Twitter

|  |  |  |  |
| --- | --- | --- | --- |
| Use case no. | 15 | | |
| Name | Link to Twitter | | |
| Description | Student links their Twitter account to the system | | |
| Version | 1.0.0 | | |
| Last edited | 05-September-2016 | | |
| Initiating actor(s) | Student | Supporting Actor(s) | Twitter authentication system |
| Entry condition(s) | The user must be logged in to a student account in the system | | |
| Exit condition(s) | The student account is updated with the OAuth token for their Twitter account in the database | | |
| Main flow | 1. The user selects link to Twitter option 2. The system passes control over to Twitter authentication system 3. The user interacts with the Twitter authentication system to complete authentication process 4. The Twitter authentication system passes control back to the system with an OAuth token 5. The system updates the user’s student account with the retrieved OAuth token in the database 6. The system informs the user that their Twitter account has been linked to their student account | | |
| Alternate flow | AF-S4: Twitter authentication system returns an error to the system   1. The system informs the user the error that has occurred during the authentication process. 2. The system exits link to Twitter use case | | |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system exits logout use case | | |

### Teacher features

#### View list of assigned student accounts

|  |  |
| --- | --- |
| Use case no. | 16 |
| Name | View list of assigned student accounts |
| Description | Teacher views a list of students assigned to them |
| Version | 1.0.0 |
| Last edited | 05-September-2016 |
| Initiating actor(s) | Teacher account |
| Entry condition(s) | The user must be logged in to a teacher account in the system |
| Exit condition(s) | - |
| Main flow | 1. The user selects view list of assigned students option 2. The system retrieves and displays the list of students assigned to the user 3. The system displays the list of students retrieved. Each student in the student list contains the following information:    1. Student ID    2. Name    3. Age    4. Depression classification |
| Alternate flow | AF-S3: The user has no student account assigned to them   1. The system displays the message “You have no student account currently assigned to you, please try to request your school administrator to assign a student to you.” 2. The system exits view list of shared student account use case |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system exits logout use case |

#### Filter students

|  |  |
| --- | --- |
| Use case no. | 17 |
| Name | Filter students |
| Description | Teacher filter students based on their depression classification |
| Version | 1.0.0 |
| Last edited | 05-September-2016 |
| Initiating actor(s) | Teacher |
| Entry condition(s) | The user must be logged in to a teacher account in the system  The teacher must have at least one student account assigned to them |
| Exit condition(s) | - |
| Main flow | 1. The user selects filter students option 2. The system displays a form with the following filter options    1. Depression classification       1. Strongly motivated       2. Motivated       3. Neutral       4. Depressed       5. Strongly depressed    2. Gender       1. Male       2. Female    3. Student name 3. The user completes one or more filter options 4. The system retrieves a list of students that matches the filter options selected 5. The system displays the list of student with each student containing the following information:    1. Student ID    2. Name    3. Age    4. Depression classification |
| Alternate flow | - |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system exits logout use case |

#### View journal entry (Teacher)

|  |  |
| --- | --- |
| Use case no. | 18 |
| Name | View journal entry (Teacher) |
| Description | Teacher views a journal entry entered by the selected student |
| Version | 1.0.0 |
| Last edited | 05-September-2016 |
| Initiating actor(s) | Teacher account |
| Entry condition(s) | The user must be logged in to a teacher account in the system  The teacher must have selected to view a student account |
| Exit condition(s) | - |
| Main flow | 1. The user selects view journal entries option 2. The system retrieves a list of journal entries submitted by the selected student 3. The system displays the list of journal entries in chronological order, with each journal entry having the following detail:    1. Date    2. Time    3. Title    4. Analyzed depression index 4. The user selects a journal entry 5. The system retrieves details of the selected journal entry 6. The system displays the journal entry with the following detail:    1. Date    2. Time    3. Title    4. Journal    5. Comments made by teacher |
| Alternate flow | - |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system exits logout use case |

#### Enter comments

|  |  |
| --- | --- |
| Use case no. | 19 |
| Name | Enter comments |
| Description | Enter comments for selected journal entry |
| Version | 1.0.0 |
| Last edited | 05-September-2016 |
| Initiating actor(s) | Teacher account |
| Entry condition(s) | The user must be logged in to a teacher account in the system  The teacher must have selected to view a journal entry  The selected journal entry must not have a comment |
| Exit condition(s) | The system creates and saves the comment record into the database |
| Main flow | 1. The user selects enter comment option 2. The system displays comment form with the following fields:    1. Comments 3. The user completes and submits the form 4. The system verifies that all fields in the form are completed 5. The system creates a comment record with detail entered in comment form along with the current date, time, journal entry identifier and user’s teacher account ID 6. The system saves the comment record into the database 7. The system informs the user that the comment has been successfully saved |
| Alternate flow | AF-S5: Some fields in the comment form are not completed   1. The system prompts the user to complete the comment form before submitting 2. The system returns to step 2 of the main flow |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system exits logout use case |

#### Search nearby counsellors

|  |  |  |  |
| --- | --- | --- | --- |
| Use case no. | 20 | | |
| Name | Search nearby counsellors | | |
| Description | Teachers search for nearby counsellors to help students with depression | | |
| Version | 1.0.0 | | |
| Last edited | 05-September-2016 | | |
| Initiating actor(s) | Teacher account | Supporting Actor(s) | Geocoder API |
| Entry condition(s) | The user must be logged in to a teacher account in the system | | |
| Exit condition(s) | - | | |
| Main flow | 1. The system displays the following options:    1. Search by GPS    2. Search by address 2. The user selects search by address option 3. The system displays an address form with the following fields:    1. Street name    2. Block number    3. Unit number    4. Postal code 4. The user completes and submits the form 5. The system verifies that the street name field in the form has been completed 6. The system queries for the address location from the geocoder API using the entered address detail 7. The system retrieves a list of counsellors sorted based on relative distance from the entered address detail to the counsellor office address 8. The system displays a list of counsellors. Each counsellor in the list will be displayed with the following information:    1. Counsellor ID    2. Name    3. Office address | | |
| Alternate flow | AF-S3: The user selects search by GPS option   1. The system retrieves the user’s current location 2. The system retrieves a list of counsellors sorted based on relative distance of their office address to the user’s current location   AF-S5: The street name field of the address form is not completed   1. The system prompts the user to complete the street name field of the address form 2. The system returns to step 3 of the main flow   AF-S6: The geocoder is unable to return an address location   1. The system informs the user that a location cannot be determined for the entered address 2. The system returns to step 2 of the main flow | | |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system exits logout use case | | |

#### View counsellor account

|  |  |
| --- | --- |
| Use case no. | 21 |
| Name | View counsellor account |
| Description | Teacher views a counsellor account |
| Version | 1.0.0 |
| Last edited | 05-September-2016 |
| Initiating actor(s) | Teacher account |
| Entry condition(s) | The user must be logged in to a teacher account in the system |
| Exit condition(s) | - |
| Main flow | 1. The user selects view counsellor account option 2. The system passes control to search nearby counsellors use case 3. The system prompts the user to select a counsellor from the displayed list 4. The user selects a counsellor account to view 5. The system displays the following counsellor account details:    1. Counsellor ID    2. Name    3. Age    4. Gender    5. Office location    6. Achievements |
| Alternate flow | - |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system exits logout use case |

#### Share student accounts

|  |  |
| --- | --- |
| Use case no. | 22 |
| Name | Share student accounts |
| Description | Teacher share student accounts with the selected counsellor |
| Version | 1.0.0 |
| Last edited | 05-September-2016 |
| Initiating actor(s) | Teacher account |
| Entry condition(s) | The user must be logged in to a teacher account in the system  The teacher must have selected to view a counsellor account  The teacher must have at least one student account assigned to them |
| Exit condition(s) | The system creates and saves a sharing record for each of the selected students and the selected counsellor into the database |
| Main flow | 1. The user selects share student accounts option 2. The system retrieves and displays a list of students assigned to the user. Each student is displayed with the following information:    1. Student ID    2. Name    3. Depression classification 3. The user selects the students to share their account information with the counsellor and submits 4. The system creates a sharing record for each selected student and the selected counsellor 5. The system saves the sharing records into the database 6. The system informs the user of the successful share |
| Alternate flow | - |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system exits logout use case |

#### Set grades for assigned students

|  |  |
| --- | --- |
| Use case no. | 23 |
| Name | Set grades for assigned students |
| Description | Teacher sets the examination grades for students assigned to them |
| Version | 1.0.0 |
| Last edited | 05-September-2016 |
| Initiating actor(s) | Teacher account |
| Entry condition(s) | The user must be logged in to a teacher account in the system |
| Exit condition(s) | The system creates and saves the grade records into the database |
| Main flow | 1. The user selects set grades option 2. The system retrieves a list of students assigned to the user 3. The system retrieves a list of modules for each student retrieved in step 2 4. The system displays a set grade form for each student with the following information:    1. Student ID    2. Name    3. A list of module the student is assigned   For every module that a retrieved student is assigned to, the following field is displayed:   1. Grade 2. The user completes the grade form and submits 3. The system displays a summary of the grades entered and prompts the user for confirmation 4. The user confirms the action 5. The system creates a grade record for each grade entered 6. The system saves the grade records into the database 7. The system informs the user the grade records has been successfully saved |
| Alternate flow | AF-S2: The user has no student account assigned to them   1. The system displays the message “You have no student account currently assigned to you, please try to request your school administrator to assign a student to you.” 2. The system exits view list of shared student account use case   AF-S7: The user rejects the action   1. The system returns to step 4 of the main flow |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system exits logout use case |

### Counsellor/Therapist features

#### View list of shared student account

|  |  |
| --- | --- |
| Use case no. | 24 |
| Name | View list of shared student account |
| Description | Counsellor views the list of student accounts shared with them |
| Version | 1.0.0 |
| Last edited | 05-September-2016 |
| Initiating actor(s) | Counsellor |
| Entry condition(s) | The user must be logged in to a counsellor account in the system |
| Exit condition(s) | - |
| Main flow | 1. The user selects view shared student accounts option 2. The system retrieves the list of student accounts shared with the user 3. The system displays the list of students retrieved. Each student account in the list will be displayed with the following information:    1. Student ID    2. Name    3. Age    4. School    5. Depression classification |
| Alternate flow | AF-S3: The user has no student account shared with them   1. The system displays the message “You have no student account information currently shared with you, please try to request the student’s account information from their respective school” 2. The system exits view list of shared student account use case |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system exits view list of shared student account use case |

#### View treatment logs

|  |  |
| --- | --- |
| Use case no. | 25 |
| Name | View treatment logs |
| Description | Counsellor views the past treatment logs of a student account |
| Version | 1.0.0 |
| Last edited | 05-September-2016 |
| Initiating actor(s) | Counsellor |
| Entry condition(s) | The user must be logged in to a counsellor account in the system  The counsellor must have selected to view a student account |
| Exit condition(s) | - |
| Main flow | 1. The user selects the view treatment logs option 2. The system retrieves a list of treatment logs saved into the system for the selected student 3. The system displays the list of treatment logs for the selected student in chronological order with each treatment logs containing the following information:    1. Treatment date    2. Treatment time    3. Name of counsellor administrating treatment    4. Type of treatment    5. Extra notes or details |
| Alternate flow | - |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system exits view treatment logs use case |

#### Enter treatment log

|  |  |
| --- | --- |
| Use case no. | 26 |
| Name | Enter treatment log |
| Description | Counsellor enters a treatment log for a treatment session into the system |
| Version | 1.0.0 |
| Last edited | 05-September-2016 |
| Initiating actor(s) | Counsellor |
| Entry condition(s) | The user must be logged in to a counsellor account in the system  The counsellor must have selected to view a student account |
| Exit condition(s) | A new treatment log is created and saved into the database |
| Main flow | 1. The user selects the enter treatment log option 2. The system displays a treatment log form with the following fields:    1. Date of the treatment    2. Time of the treatment    3. Type of treatment    4. Extra notes or details specific to the treatment 3. The user completes the treatment log form 4. The user submits the form 5. The system verifies that all fields are completed 6. The system prompts the user for confirmation to save treatment log 7. The user confirms to save treatment log 8. The system creates a treatment log with the details entered in the treatment log form, along with the selected student ID and user’s counsellor ID 9. The system saves the treatment log into the database 10. The system informs the user that the treatment log has been successfully saved |
| Alternate flow | AF-S5: Some fields in the treatment log form are not completed   1. The system prompts the user to complete the treatment log form before submitting 2. The system returns to step 2 of the main flow   AS-S7: The user rejects saving treatment log action   1. The system returns to step 2 of the main flow |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system returns to step 2 of the main flow |

#### View journal entry (Counsellor)

|  |  |
| --- | --- |
| Use case no. | 27 |
| Name | View journal entry (Counsellor) |
| Description | Counsellor views a journal entry entered by the selected student |
| Version | 1.0.0 |
| Last edited | 05-September-2016 |
| Initiating actor(s) | Counsellor |
| Entry condition(s) | The user must be logged in to a counsellor account in the system  The counsellor must have selected to view a student account |
| Exit condition(s) | - |
| Main flow | 1. The user selects view journal entries option 2. The system retrieves a list of journal entries submitted by the selected student 3. The system displays the list of journal entries in chronological order, with each journal entry having the following detail:    1. Date    2. Time    3. Title    4. Analyzed depression index 4. The user selects a journal entry 5. The system retrieves details of the selected journal entry 6. The system displays the journal entry with the following detail:    1. Date    2. Time    3. Analyzed depression index    4. Title    5. Journal    6. Comments made by teacher |
| Alternate flow | - |
| Exception(s) | EX1: The server is not available to process the user request   1. The system displays the message “Server is busy, please try again later!” 2. The system exits view journal entry use case |

## Sequence diagram

# Conclusion

Depression is a mental disorder that can lead to many disastrous consequences. Early treatment can potentially prevent the onset of suicidal thoughts for the depression sufferers. Our proposed solution can greatly increase the effectiveness of early detection of depression symptoms and aid students in receiving appropriate treatments in the early stages. Early detection and treatment is of paramount importance as it could mean the difference between life and death of the depression sufferers.

# References

[1]

Statistics retrieved from page 4 line 7 of the following paper: <https://www.moh.gov.sg/content/dam/moh_web/HPP/Doctors/cpg_medical/current/2012/depression/Depression%20CPG_R11.pdf>