

SW Engineering CSC648/848

Section-04 Fall 2024

Team 01

Project Title :

“ChillMate”

Eiffel Valentino (Team Leader)
Luis Carrillo (GitHub Master)
Gio Jung (Scrum Master)
Chun Kai Liu (FrontEnd Leader)
William Widjaja (BackEnd Leader)
Sneha katturu (FrontEnd)
Jay Lodha (BackEnd)

Milestone 1

Date Submitted	Oct. 04. 2024
Date Revised	

1. Executive Summary

Mental health and well-being are critical challenges in university settings, with students and faculty facing academic pressure, career uncertainties, and school-life balance issues. This mounting stress often leads to anxiety, procrastination, and a lack of focus, which can hinder both academic and personal success. Unfortunately, traditional mental health resources are often underutilized due to accessibility barriers, long wait times, and the stigma associated with seeking help.

ChillMate is designed to bridge this gap by offering a digital, stress-focused support tool for SFSU students and faculty. The core of the application is a chatbot that provides guidance on school-related stress, including academic burnout, procrastination, and career anxiety. It directs users to university-specific resources such as counseling services, workshops, and academic support centers, while ensuring ethical boundaries by not diagnosing mental health conditions or providing medical advice.

What sets ChillMate apart is its targeted focus on school-related stress, making it more relevant than generic mental health apps. By personalizing the experience based on the user's concerns, the chatbot provides timely, appropriate guidance, ensuring that users feel supported without feeling overwhelmed. By limiting its scope to university-specific stress, ChillMate maintains a clear focus, directing users to relevant resources while avoiding extraneous or inappropriate advice.

Our team consists of passionate SFSU students from diverse fields, including computer science, psychology, and education. With our personal understanding of the unique stressors within a university environment, we have created an intuitive and empathetic platform. We believe that ChillMate will enhance the mental well-being of the campus community, fostering a healthier, more supportive academic environment. Backing this project means supporting a scalable, innovative approach to mental health that addresses a growing need, starting with SFSU.

2. Personas and User Stories:

Persona 1: Alex the Undergraduate Student

Age: 21

Occupation: College Student (Psychology Major)

Location: Village At Centennial Square - SFSU



Background:

Alex is a junior in college, juggling a part-time job, academics, and a social life. They experience anxiety, especially during exams, and often feel overwhelmed by the pressure to perform.

Goals:

Manage anxiety and stress effectively.

Access mental health resources easily.

Have someone to talk to about sensitive topics safely

Story:

As finals approach, Alex feels the weight of academic pressure mounting. They hear about the mental health app during an orientation session. Intrigued, they download it and explore its features. The mood tracker helps Alex monitor their anxiety levels. The chatbot was able to offer a variety of resources to help Alex with his breathing since he was experiencing difficulty with it. By the end of the semester, Alex feels more equipped to handle stress and appreciates having a community that understands their challenges.

Persona 2: Dr. Michelle Torres the Faculty Member

Age: 45

Occupation: Associate Professor of Biology

Location: Ingleside, San Francisco

Background:

Dr. Torres is passionate about teaching but often feels overwhelmed by administrative tasks and the emotional needs of her students. She wants to support her students' mental health while managing her own stress.

Goals:

Enhance her ability to support students in distress.

Access resources on mental health for faculty.

Reduce her own stress levels.

Story:

After noticing a rise in anxiety and stress among her students, Dr. Torres seeks ways to be a better support. She learns about the mental health app during a faculty meeting and decides to explore it. The app offers faculty-specific resources, including articles on recognizing signs



of distress and strategies for effective communication. For her administrative class the chatbot was able to give some tips with time and task management. This encourages her to create anonymous forms as a way to reach out to her students and for her to offer help. By the end of the semester, she feels more confident in her role as a mentor and notices improved engagement in her classroom.

Persona 3: Sam the Graduate Student

Age: 27

Occupation: Graduate Student (Masters in Education)

Location: Oceanview, San Francisco

Background:

Sam is pursuing a master's degree while working as a teaching assistant. Balancing coursework, teaching duties, and personal life has taken a toll on their mental health, leading to feelings of burnout.

Goals:

Develop effective coping mechanisms for stress.
Access academic resources related to mental health.
Build connections with faculty and fellow students.

Story:

Feeling overwhelmed and burned out, Sam hears about the mental health app through a friend in the program. They download it and are relieved to find resources specifically for graduate students. The app offers information about workshops on time management and stress reduction techniques, which Sam finds particularly helpful. Additionally, Sam uses the forum to connect with fellow TAs, sharing experiences and strategies for balancing their roles. By integrating the app into their routine, Sam finds a renewed sense of balance and purpose.



3. Data Definitions :

User - entity that stores the information of registered accounts

- firstName: First name of user
- lastName: Last name of user
- email: Email address of user. Used for login
- password: Password of user. Used for login, and will be hashed
- Address: home address of user
- phone number: phone number of user

Chat Interaction - the dialog exchanged between the student and the chatbot.

- message: Content provided by the user
- response: Content provided by the chatbot
- timestamp: Time of the interaction
- sentimentAnalysis: A score or indicator representing the emotional tone of the student's message

Resources - entity that provide resource to users

- Recourse Title : Title of the resource
- Recourse Link : Linke of the resource
- Recourse Body : Details of the resource
- SFSU Specific (Yes or NO) : Is the resource SFSU Specific?

Mental Health Entry - entity that stores the user current health and emotional status

- mood: The current mood of the user (e.g., Happy, Sad, Anxious, Stressed)
- timestamp: The time the entry was made
- mentalHealthScore: A numerical or categorical score based on the student's inputs (e.g., stress or anxiety levels)

Goal/Task - entity that stores the user goal / tasks that they want to achieve

- Goal: the goal that user input
- User: which user wants to achieve this goal
- Type: whether the goal will be recurring or just one time
- Status: tracks whether the goal have been accomplished or not

Journal Entry - entity that stores the journal entry of the user

- content: journal entry based on the user input
- timestamp: Time the journal entry was created
- sentimentTag: Optional sentiment analysis of the content to assess emotional undertones

Reminder/Notification - alert sent to the user to remind them to interact with the app or prompt a wellness check-in

- type: Can be a reminder (e.g., for mood check-in), or an alert (e.g., if mental health score indicates high stress)
- timestamp: When the notification was sent
- trigger: The condition that caused the notification (e.g., absence of interaction for a certain period)

Data and Analytics - data that has been stripped of personally identifiable information, used for analytics and reporting. Also to monitor app performance and review aggregate mental health data

- moodTrends: Aggregated data without identifiable information
- usageMetrics: General usage patterns like frequency of app interaction
- usageAnalytics: Metrics about app usage

Forum - entity that will store the posts and comments that will be created in the forum

- Text: contains the user input to the forum
- Topic: determines what topic this is about, whether advice, tips, question etc
- Type: determines whether the user input is a comment or a post

4. List of Functional Requirements :

- Creating an account, and being able to log in to the account
- Users have their own profile
- Users can have a conversation with chatbot wherein the chatbot supports features such as:
 1. Resource Recommendation: The chatbot could provide personalized resources like articles, videos, or links to campus services based on the user's issue, such as study tips, mental health resources, or academic support.
 2. Goal Setting & Tracking: The bot can help students set goals (e.g., academic, fitness, personal) and track progress. It could give friendly reminders and motivation to encourage goal completion.
 3. Crisis Response and Escalation: If the chatbot detects signs of distress or crisis, it should be able to escalate the situation by directing the user to a counselor, or provide emergency contacts. We will basically track the user's mood/ tone during the conversation.
 4. Conversational Context Retention: Store conversation context so the chatbot remembers past interactions, allowing it to provide better continuity and personalized suggestions in future sessions.
 5. Campus-Specific Information: Offer location-specific details like event reminders, campus maps, office hours for different departments, or FAQs about student services. This can make the bot even more valuable for college students.
 6. Survey and Feedback Collection: Periodically ask students about their satisfaction with various aspects of campus life, allowing the institution to gather data on student well-being and experiences
 7. Conversational Mood Journal: Allow students to record how they're feeling each day in a conversational style. The chatbot can track this over time and give them insights or encourage them when it notices patterns of stress.
- Provide a journal area where user can use write their problem
- Provide resources for user problems
 - Users should be able to save the resources for future uses
- Give a daily word of encouragement
- A forum where user can post their problems, tips, solutions
 - User can also comment on each other's post
 - A report feature is also provided to report inappropriate posts
- Users can create a Task list on the application based on the checklist provided by the chatbot (as part of the Goal Setting & Tracking feature from chatbot).

5. List of Non-Functional Requirements :

- Performance: The app should be reactive and fast-responsive to ensure a smooth and responsive user experience.
- Security: User data should always remain anonymous. The journal that the user writes should also not be able to be accessed.
- Capability: The website should work on all devices both phone and laptop. The app should also work on all Operating systems.
- App Updates: The app should have an automatic system update that will always provide the updated version.
- Storage Space: User interaction history should be stored for a maximum of 6 months before automatic deletion.
- Usability: The chatbot interface should have an intuitive and easy-to-navigate user experience with a simple UI accessible on both desktop and mobile devices.
- Fault Tolerance: Auto-retry mechanisms should be in place to handle temporary service interruptions, ensuring a seamless user experience.
- Scalability: It should support integration with external systems like university counseling services without requiring significant reconfiguration.
- Maintainability: The system's architecture should allow for regular updates and feature enhancements without downtime, ensuring easy maintainability and future development.
- Backup and Disaster Recovery: The system should be able to recover within 1 hour in case of catastrophic failures.
- User Feedback Mechanism: The chatbot should allow users to provide feedback about responses and service quality, with feedback loops to improve the chatbot's performance and responsiveness over time.

6. Competitive Analysis :

Features	Headspace	Sanvello	CampusWell	ChillMate
Mood Tracker	Yes	Yes	No	Yes
Forum/Community support	No	Yes	Yes	Yes
Tailored for SFSU	No	No	No	Yes
Educational Resources	Yes	Yes	No	Yes
Guided Meditation/Relaxation	Yes	Yes	Yes	Yes

Compared to existing solutions like Headspace, Sanvello, and CampusWell, our planned web application, ChillMate, stands out by offering SFSU-specific tailored content, which directly addresses the unique academic and mental health concerns of the SFSU community. While competitors provide mood tracking and forums, they are often generic, whereas our platform integrates these features into a context relevant to students. This makes our solution more personalized and community-focused, ensuring that users can access resources, track their moods, and connect with peers facing similar challenges, all while receiving SFSU-related information and support.

7. High-Level System Requirements :

- Server Host: AWS EC2
- Operating System: Ubuntu 24.04
- Database: MongoDB (Atlas Free Tier)
- Web Server: Vercel
- Front-end Framework: React
- Server-Side Language: Python
- Web Application Framework: Flask
- Additional Technologies: FastAPI
- IDE: Visual Studio Code

8. Team: list student names, name of the roles for each member. *If you form the study group, please list them too with their key milestones. (If you present detailed study plan, you will earn extra points).*

Eiffel Valentino (Team Leader)
Luis Carrillo (Github Master)
Gio Jung (Scrum Master)
Chun Kai Liu (FrontEnd Leader)
William Widjaja (BackEnd Leader)
Sneha katturu (FrontEnd)
Jay Lodha (BackEnd)

Study Plan

Week 1

Subject: AWS

Leader: Gio Jung (4)

Students: Eiffel (3), Luis (2), William (1), Kevin (2), Jay (2), Sneha (2)

Details: The focus will be on AWS basics, including setting up EC2 instances, understanding the AWS dashboard, and learning how to integrate AWS services with other tools..

Week 2

Subject: Ubuntu

Leader: Eiffel Valentino (5)

Students: Luis (3), William (1), Kevin (3), Jay (3), Gio (4), Sneha (1)

Details: Learn Ubuntu basics, such as package management, terminal commands, and system configurations.

Week 3

Subject: MongoDB

Leader: Jay Lodha (4)

Students: Eiffel (2), Luis (1), William (3), Kevin (2), Gio (1), Sneha (2)

Details: Cover MongoDB CRUD operations and connect it to backend systems.

Week 4

Subject: Vercel

Leader: Eiffel Valentino (5)

Students: Luis (2), William (2), Kevin (2), Jay (1), Gio (1), Sneha (1)

Details: Focus on deploying applications on Vercel and utilizing its deployment features.

Week 5

Subject: React

Leader: William Widjaja (3)

Students: Eiffel (2), Luis (2), Kevin (3), Jay (1), Gio (1), Sneha (2)

Details: Learn to create React components and manage state for dynamic web applications.

Week 6

Subject: Python

Leader: Jay Lodha (5)

Students: Eiffel (3), Luis (1), William (4), Kevin (3), Gio (3), Sneha (2)

Details: Learn Python fundamentals such as syntax, data structures, and object-oriented programming. Also, focus on the server-side implementation.

Week 7

Subject: Flask

Leader: Jay Lodha (5)

Students: Eiffel (3), Luis (1), William (4), Kevin (3), Gio (3), Sneha (2)

Details: Learn to set up routes, handle HTTP requests, and return responses and to integrate with MongoDB.

Week 8

Subject: VSCode

Leader: Eiffel Valentino (5)

Students: Luis (3), William (1), Kevin (3), Jay (4), Gio (3), Sneha (3)

Details: Explore advanced features in VSCode, such as debugging, extensions, and remote collaboration that will be used to manage the server.

9. Checklist: *for each below item, you must answer with only one of the following: **DONE**; or **ON TRACK** (meaning it will be done on time, and no issues perceived); or **ISSUE** (you have some problems, and then define what is the problem with 1-3 lines)*

- ☒ Team found a time slot to meet outside of the class – DONE
- ☒ Scrum Master shares meeting minutes with everyone after each meeting – DONE
- ☒ GitHub Master chosen – DONE
- ☒ Everyone sets up their local development environment from the team's git repo. – DONE
- ☒ Team decided and agreed together on using the listed SW tools and deployment server – DONE
- ☒ Team lead ensured that all team members read the final M1 and agree/understand it before submission – DONE
- ☐ Team ready and able to use the chosen back/front-end frameworks. – ON TRACK