**MentalHelp**

**Hayden Guo, Remy Xiong, Jiebin Lee**

# PROJECT PROBLEM STATEMENT

Many people nowadays struggle to keep track of and understand their mental well-being. The program aims to help users record daily emotional states and behaviors and analyze patterns.

# CONCISE DESIGN OVERVIEW

We will develop a cross-platform desktop application in Java using JavaFX for the user interface. Users can select any date from a simple calendar view for the current week and log their mood (happiness, sadness, stress). After completing entries for all days, the system generates graphs of their mood throughout the week and comparing it to the previous week.

# SCOPE STATEMENT

Objective: To provide a working, accessible application for anyone (with a computer) to use and to support any mental health complications they might have.

In Scope:

* Desktop UI supporting Windows, macOS, and Linux (using JavaFX)
* Calendar widget for selecting days of the current week
* Daily entry: mood rating
* Weekly summary report with charts
* Simple local data storage (probably SQLite)

Out of Scope:

* User authentication or multi-user support
* Cloud sync or remote storage
* Notifications/reminders
* Advanced analytics (like using machine learning)
* Mobile or web support

# TIMELINE

|  |  |
| --- | --- |
| Do by end of: | Task: |
| 5/23 | Review scope/requirements, confirm GitHub working correctly, assign team roles |
| 5/28 | Sketch designs for main screens and define classes, relationships, and how data will be saved. |
| 5/29 | Set up a bare bones prototype of the application. |
| 5/30 | Implement the calendar widget. |
| 6/2 | Implement the daily entry form. |
| 6/4 | Configure data storage. |
| 6/5 | Implement weekly summaries. |
| 6/6 | Add charts to weekly summaries. |
| 6/9 | Add recommendations/resources to weekly summaries. |
| 6/11 | Connect everything together. |
| 6/12 | Start testing. |
| 6/13 | Polishing, bugfixing, etc. |
| 6/16 | Finish documentation and presentation. |

# TEST PLAN

* Verify each component works correctly and matches with our scope, design overview, etc.
* Send program to multiple other people and get their feedback and experience
* Continue to use program until project is due to ensure functionality

# RISKS AND CONTINGENCY PLANS

|  |  |  |
| --- | --- | --- |
| Risk | Likelihood | Mitigation |
| Team member availability | Medium | Assign overlapping tasks and maintain teamwork. |
| Scope creep | High | Require any new changes to be thoroughly reviewed by the entire team and prioritized last. |
| Java learning curve | High | Pair programming, simplify program |
| Data loss/corruption | Low | Implement backups |
| Project takes longer than expected | High | Simplify project and delay non-essential features |