Project Proposal for CSCD 350 Spring 2024

TimeTracker

1.0

Team 4: Ghidra

Submitted By

Allen Chang achang3@ewu.edu
Carrie Sargent csargent3@ewu.edu
Earl Quinto equinto@ewu.edu
Ethan Crawford ecrawford4@ewu.edu
William Kern wkern1@ewu.edu

Instructor: Sanmeet Kaur
GSA: Dominic Maclsaac, Harley Davis
Lab Section: Section 1
Date: <place the date of submission here>
GitHub Repository: https://github.com/Sanmeet-EWU/github-teams-project-bid-ghidra.git

1. Motivation

TimeTracker is a web application designed to enhance users' event planning experience through its intuitive calendar interface. What separates TimeTracker from the alternative is the it's precision enabling users to schedule events down to the minute. This level of detail ensures that the event is carefully considered and accounted for in the planning process. Moreover, TimeTracker caters to events that span across several days, providing a comprehensive solution for tracking and managing long-term projects or activities.

1.1 Issues with other products

TimeTracker hopes to improve upon the limitations presented by existing calendar applications, such as Google Calendar or Outlook Calendar, to improve the overall quality of life for the users. One primary motivation behind TimeTracker's development is the recognition that traditional calendar applications often lack the precision and granularity required for meticulous planning. TimeTracker fills this gap by allowing users to plan events with unparalleled precision, ensuring that every minute is accounted for and optimized. In addition to precision planning, TimeTracker's ability to plan events that spans across multiple days addresses several challenges faced by users in managing long-term projects and activities. Also, TimeTracker provides users the ability to duplicate existing schedules to streamline the process of managing recurring tasks, events, or activities across different days, ensuring consistency and saving time while scheduling and planning. Feature also facilitates the integration of daily routines into the users' schedule.

1.2 Audience

TimeTracker's target audience comprises individuals who prioritizes precise planning down to the minute and values the convenience of duplicating schedules across different days. Additionally, TimeTracker appeals to users managing or participating in events that span several days or weeks in the future, offering a comprehensive tool for organizing and monitoring extended timelines. By catering to both target audiences, TimeTracker provides a versatile solution for optimizing users time management.

Our users may be busy individuals that would like to plan out their entire day to the minute to stay organized and productive. The user may also have multiple calendar applications with different sets of events, some of which could conflict with time slots or overlap in a confusing way. Our users have this problem because of the way that popular calendar apps schedule events. For example, Google Calendar will automatically add plane tickets or hotel reservations as events if they are purchased through a Google account. While this

automatic scheduling can be helpful, the events that are added to the calendar often overlap times and lack specificity. Notifications for one event may be overtaken by other events that occur, and thus be forgotten. Many major calendar apps, such as Google, Microsoft, and Apple struggle with this issue.

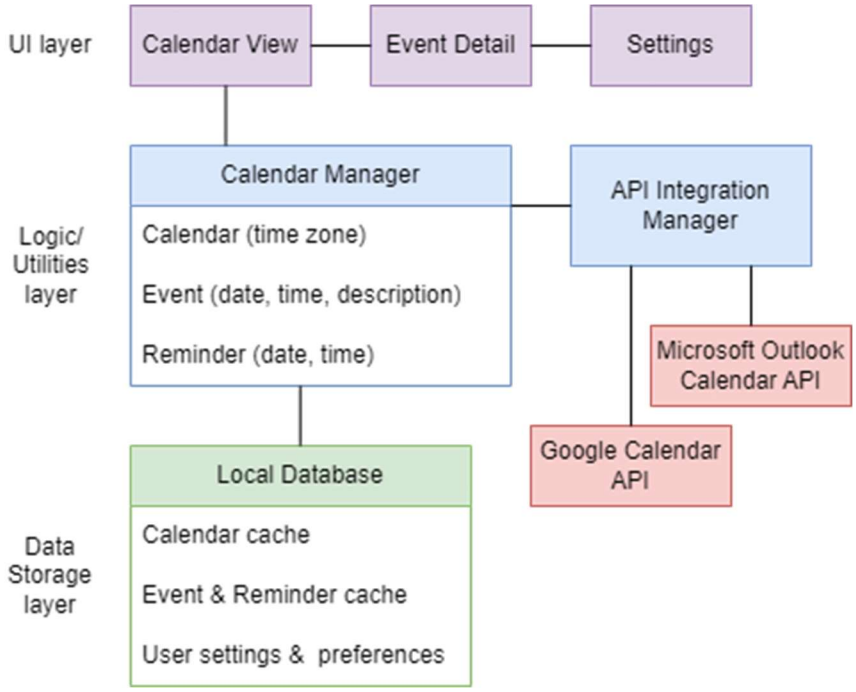
1.3 How Our Product Answers Audience Needs

With TimeTracker, these users can log information that is much more specific than other calendar apps. Timeframes for events can be planned down to the minute as a block spanning its entire duration. Time blocks can stack seamlessly, allowing users to easily distinguish when events start and end, as well as which events are ongoing while not actively needing attention. This allows for a seamless user experience which streamlines the information logjam that many calendar applications create.

2. Approach

2.1 Framework

To implement the desired features in TimeTracker, our Development team has opted to leverage JavaFX and CalendarFX. JavaFX provides a robust framework for creating user-friendly graphic interfaces, while CalendarFX specializes in calendar-related functionalities, making it an ideal choice for TimeTracker's calendar application. By utilizing these open-source platforms and libraries, the team lowers the cost for the development while also benefiting from the resources and tutorials available to support our implementation. We can see with the diagram how the general structure of TimeTracker would look like.



2.2 Development

The Development environment for TimeTracker will primarily require Java programming expertise, as Java is the primary programming language for JavaFX and CalendarFX. Additionally, Developers will need access to development tools such as an IDE like Eclipse for debugging. In terms of software dependencies, the team will integrate JavaFX and CalendarFX libraries into the development environment to access their respective functionalities. This includes user interface components, event handling mechanisms, data binding capabilities, and calendar specific features like recurring event management and date navigation.

2.3 Database Systems

Furthermore, data storage will be localized to individual users' devices, requiring the implementation of file handling and storage management within the application. This aspect of development may require additional libraries or APIs depending on the platform and file system requirements.

The implementation using JavaFX and CalendarFX streamlines the development process and enables resources to the team for the development of a user focused calendar scheduling and planning application.

3. Risk

When it comes to developing TimeTracker, there are several potential roadblocks that could hinder our progress. These risks primarily stem from

technical complexities, resource constraints, and setbacks during the development process.

3.1 System Incompatibility

One risk is encountering compatibility issues between different systems used in TimeTracker. For example, our app might not work properly on certain devices or may cause ‘glitch’ in certain areas. There could be conflicts between different software components.

3.2 Running out of time

Another risk is not having enough time to complete all the tasks required to build the TimeTracker. This could happen due to underestimating the time needed for certain features or facing unexpected delays along the way.

3.3 Resource Constraints

Some of us haven’t developed an application like this before, and having a limited knowledge about may cause struggles to meet our development goals and deliver a high-quality product on schedule.

4. Challenges

Building a TimeTracker comes with its share of tough tasks. First, we must make the app easy for everyone to use. That means designing TimeTracker so that even users that are not tech-savvy can figure it out without any trouble. There is also the issue of keeping people's data safe. We have to make sure that all the personal info stored in TimeTracker stays private and secure. Additionally, as more and more people start using TimeTracker, we need to make sure it keeps working smoothly for everyone, no matter how many users we have.

4.1 Schedule

Our project schedule lays out the steps we will take to build TimeTracker. We start by planning what we need to do, then move on to designing how TimeTracker will look and work. After that, we will start coding and testing, making sure the basics work smoothly,then start adding more. Once TimeTracker is ready, we will deploy it for 2-3 users to use and give us thorough feedback. Throughout the process, we will keep everyone informed and make sure we stay on track to deliver a great product on time.

4.2 Mitigation of risk

To address the challenges and risks in developing TimeTracker, we have come up with a solid plan. In response to potential compatibility issues between different systems utilized in TimeTracker, we will test it a lot to make sure it works on every system. Additionally, we'll be careful with our time so we don't fall behind schedule. We will break our work into small tasks, set realistic deadlines, and keep a close eye on our progress to make sure we're on track.

4.3 How to measure success

Measuring success with TimeTracker is all about checking if it does what we wanted and if users like it. We will see if it has all the features we planned and if it works well. By asking users what they think and listening to their feedback, we will keep improving the app and we will also keep an eye on technical difficulties, like how often it crashes or if it's slow. Overall, success means this app would make people's lives easier and help them get more done.