Project Title: ClockWork

Team Members:

- Anthony Menendez amenendezmen2022@my.fit.edu
- Christian Ott cott2020@my.fit.edu
- Pierson Hendricks phendricks2023@my.fit.edu
- Peter Stelzer pstelzer2023@my.fit.edu

Faculty Advisor: Dr. David Luginbuhl dluginbuhl@fit.edu

Client: Dr. David Luginbuhl CSE Professor & Faculty Advisor

Meeting Dates with Client:

- Meeting 1: Aug 22, 2025

Goal and Motivation: Assist in the estimation of task time cost

Students often have difficulty with time management when faced with multiple tasks across school, work, etc. This is in part due to a poor ability to estimate how much time a certain task will take. Our project seeks to alleviate this by providing a timer students can use to clock how much time they spend on a given task after providing the app with an estimated time of completion. They will then receive feedback based on their estimate and the actual time taken, which is designed to help them develop their ability to estimate the time needed to complete a task and, therefore, improve their time management skills.

Approach (key features of the system):

- Users can track their work habits
    - User can register tasks to track
        1. Users can choose from preconfigured task profiles like "MathLab Homework", "Weekly History Essay", "Morning Routine", etc.
        2. Users can define their own task profiles specific to their needs
    - Users can begin, pause, and end sessions of their registered tasks where they provide a time-cost prediction and time themself as they complete the task
    - Users can view their complete task session history to see how much time they spend working
    - Users can view task session timelines to compare how much time they spent working compared to taking breaks
- Users can refine their ability to estimate the time they require to complete tasks
    - Users can make predictions about how long a task will take to complete
    - Users can compare their predicted time with their actual time

    - Users can track the accuracy of their estimations over time for all of their registered tasks
    - Users can specify task parameters, such as difficulty or category, to link similar tasks together
    - Users can view reports detailing which task characteristics affect their estimation accuracy
    - For instance, a STEM student might create a Calc Homework profile for their weekly math practice problems. At the start of each session they will make an estimate, and at the end of each session they will compare their performance to the estimate. They will notice that they usually underestimate this profile by 30% and begin to adjust their estimates to account for that. As they use the app to track more of their tasks, the performance reports will inform them that they tend to overestimate reportedly low difficulty tasks by 10% and underestimate reportedly high difficulty tasks by 60%.
- Users can view time-cost predictions from the app for inputted task sessions
    - Users can view an app generated range of time-cost predictions that adapt to the user's performance and supplied task parameters

Novel features/functionalities:

The main novel features of the app are those that realize its focus on empowering the user to improve their time estimation performance. These include prompting the user to make an estimate before they begin a task, the post-mortem report that shows their estimate's accuracy, and the macro reports that show their performance over time. These features have not been implemented in other applications, leaving the niche of time estimation open for our application. The timing of task completion is used as a measure of user performance in business management to monitor efficiency. Our application aims to improve user's time estimation capabilities, thereby allowing them to work more efficiently.

Algorithms and tools: potentially useful algorithms and software tools

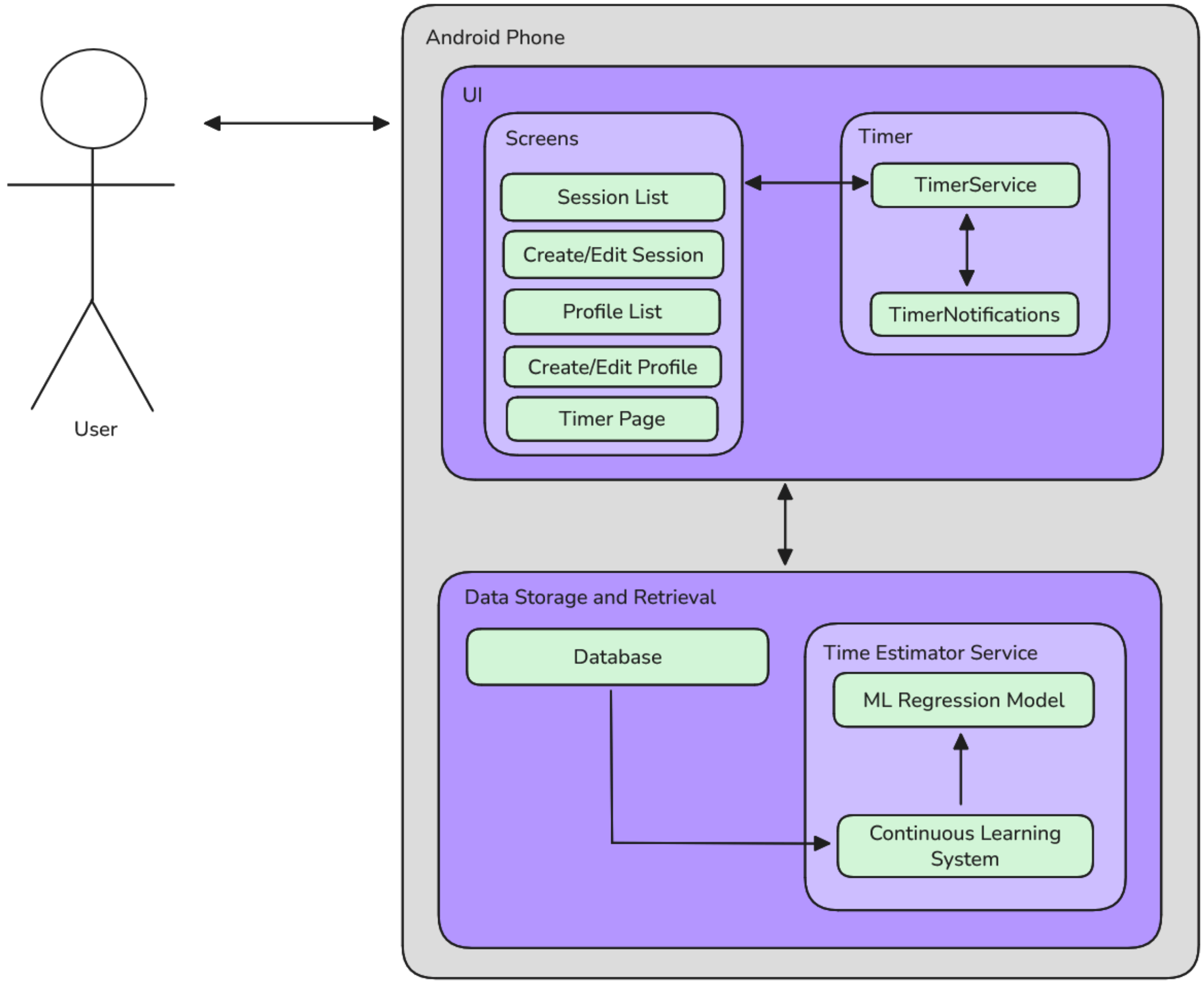
- Kotlin
    - Preferred language for Android App development.
- Android Studio
    - Mobile (Android focused) development environment.
    - Software emulation of Android devices for app design and development.
    - Android Room tools
- Jetpack Compose
    - Modern Android UI library and ecosystem
- Android Room
    - Kotlin/Jetpack compose SQLite database implementation

- LiteRT
    - Google’s solution to running ML models on mobile devices
- XGBoost
    - Lightweight tree-based ML model that works well with diverse parameters
- SARIMAX
    - Classic statistical time-series model for forecasting with seasonal dependence
- Bayesian Regression
    - Probabilistic model that uses Bayes’ Theorem to produce a distribution
    - Can be used in tandem with either XGBoost or SARIMAX to generate error spread

Technical Challenges: Discuss three main CSE-related challenges.

- Implementing automatic time cost estimation using machine learning models on mobile device hardware.
- Interpreting user performance and programmatically formulating progress reports
- Visualization of session execution timeline

Design



Evaluation

Convenience/Ergonomics: The success of the project is predicated on how convenient the tool is to use. Users might find that by using the app they have greatly improved their ability to judge how long they require to complete tasks or that by using the app they have improved their work habits; however, if the app is not easy to adopt into their work routine or if users find it too annoying to use the app as required to achieve those results. We intend to measure this through a user survey. We will find a group of people (hopefully around ten) to use the app for a few weeks and afterward ask them how often they used the app to track their completed tasks and for how many tasks they first provided an estimate and then considered the accuracy of their estimate after the task was finished. If the user names any instances of eschewing task tracking or forgoing providing an initial estimate, we will ask what factors caused it.

Effectiveness: The success of the project is predicated on how effective the app is as a tool to improve the user's estimation skills and work habits. If users dutifully used the app as intended but saw no improvement in their estimation skills or work habits, then the project has failed its goal. We intend to measure this through a user survey. We will find a group of people (hopefully around ten) to use the app for a few weeks and afterward ask them how they felt the app altered their estimation skills and work habits. We will also request that the users share their statistics, which will show us how their estimates changed over time compared to their actual times.

Accuracy: A very simple metric to determine success comes from the more central portion of our application, the automatic time-estimation. Users will be offered an estimated completion time utilizing their previous statistics and the new task's parameters. We can judge the accuracy by simply comparing the system's estimate to what the actual result ends up being. We can also observe how well this system adjusts over time to gauge if the system becomes more accurate with more data.

Progress Summary

Module/Feature	**Completion %**	**Todo**
Session Timer	85%	● Finish live notifications ● Always accessible in-app controls ● Ensure resilience
Task Profiles	0%	Not started
Task Sessions	65%	● Association with profiles ● Association with categories ● Save user estimate and auto estimate ● Timeline markers ● Completed sessions ● Delete session
User History & Statistics	0%	Not started
Auto Timecost Estimate	0%	Not started
Task Categories	0%	Not started

Milestone 4 (Sep 29): itemized tasks:

- Add task session markers
- Save task session user estimate

- Task profiles
- Completed sessions
- Finish live timer notification
- Implement first iteration auto timecost estimator (single model, based solely on user estimate)
- Task session deletion

Milestone 5 (Oct 27): itemized tasks:

- Implement second iteration auto timecost estimator (incorporates user parameters or Bayesian Regression or both)
- Session timeline visualization and editing
- Task Categories
- User history and statistics
- Conduct evaluation and analyze results
- Create poster for Senior Design Showcase

Milestone 6 (Nov 24): itemized tasks:

- Polish and unify user interface design
- Task profile name template variables
- Test/demo of the entire system
- Conduct evaluation and analyze results
- Create user/developer manual
- Create demo video

Task Matrix for Milestone 4:

Task	Anthony	Christian	Peter	Pierson
1. Task session markers	0%	0%	50%	50%
2. Save task session user estimate	0%	0%	20%	80%
3. Task profiles	0%	0%	100%	0%
4. Completed sessions	0%	100%	0%	0%

5. Finish live timer notification	0%	0%	50%	50%
6. 1st iteration auto estimator	75%	0%	25%	0%
7. Task session deletion	0%	0%	0%	100%

Description (at least a few sentences) of each planned task for Milestone 4:

Task 1: One of the system requirements is to enable the user to place markers during task execution which will appear on the timeline once the task is completed. The objective for this milestone task is to implement creating these markers while the timer is running and saving them to app storage. This should be a trivial task since most of this functionality is already present in segment serialization.

Task 2: As of writing, when the user creates or edits a new task session, they may specify an estimate. This estimate, however, is not saved. The objective for this milestone task is to implement the saving of the user estimate. The task session edit form should also be enhanced to allow the user to forgo specifying an estimate and specify one if they had previously omitted it.

Task 3: One of the system requirements is to enable the user to define task profiles which act as templates and aggregators for task sessions created with them. The objective for this milestone task is to enhance the database schema to support profile serialization, add a UI page that lists all profiles, add a UI page for profile creation, add a UI page for profile editing, add the ability to delete a profile, and add a UI page that lists all of the sessions made with the profile.

Task 4: As of writing, users can create tasks, start them, but they cannot complete them. The objective of this milestone task is to enable users to complete tasks. This involves adding a procedure to safely disengage the session from the timer, close all loaded instances of the session, and update its record in the database; adding a UI page for the session's post-mortem report, and adding a list of all completed tasks.

Task 5: As of writing, the timer posts a system notification which displays the elapsed work time and break time of the session and has a button to resume or pause the session depending on its state. The notification disappears when the timer is suspended. The objective for this milestone is to make the notification more substantial with more controls and a more refined presentation including content formatting, higher priority in the notification center, and visibility on the lock screen.

Task 6: One of the system requirements stipulates that the system shall generate its own timecost estimation based on user-specified parameters and performance history. After researching the problem, it seems that the solution that has the best chance of yielding quality results is a machine learning model. The objective of this milestone task is to implement an auto time estimator service that uses either an XGBoost or SARIMAX model with user estimate as input and an adjusted estimate as output that retrains itself after a set number of new datapoints.

Task 7: As of writing, the user can create and edit task sessions, but they cannot delete them once created. The objective of this milestone task is to implement the ability to delete a task session from the UI.

Approval from Faculty Advisor:

- I have discussed with the team and approved this project plan. I will evaluate the progress and assign a grade for each of the three milestones.
- Signature: ________________________________ Date: ________