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

Progress of Current Milestone:

Task	Completion %	Anthony	Christian	Peter	Pierson	To do
1. Finish session deletion	100%	0%	0%	50%	50%	
2. Finish profile deletion	100%	0%	0%	100%	0%	
3. Completed session timeline visualization	45%	0%	0%	0%	100%	Frontend and completion
4. Completed session execution event editing	0%	-	-	-	-	Descoped
5. User history and statistics page	60%	0%	100%	0%	0%	Template performance statistics
6. Custom profile fields	0%	-	-	-	-	Descoped
7. Next auto estimate iteration	0%	-	-	-	-	Descoped
8. Conduct evaluation and analyze results	100%	0%	0%	100%	0%	

9. Create poster for Senior Design Showcase	100%	50%	0%	50%	0%	
10. Fix timer notification sync issues	100%*	0%	0%	100%	0%	*Was not able to trigger the failure
11. Fix Android 10 crash on open	100%	0%	0%	100%	0%	
12. Fix visuals breaking from greater than default system font scale and large strings	90%	20%	0%	80%	0%	Template session list session UI items and Template name
13. Combine worktime and breaktime during task execution	90%	0%	0%	100%	0%	Template session list completed session list UI items
14. Make fine adjustments to selected estimate easier	100%	0%	0%	100%	0%	
15. Make completed task properties editable	100%	0%	0%	100%	0%	
16. Rename task profiles to task templates	100%	100%	0%	0%	0%	

Discussion of Current Milestone Tasks:

- Task 1: This is a left-over task from the last milestone. Users should be able to delete any task (including completed tasks) so that they can fix mistakes. This involved declaring database interface function signatures for session record deletion, then a corresponding function in the session repository. A submenu was created on the Timer Page to house the button to trigger the action. Before this change, the app required that a record exists when it is requested from the database by id and continues to exist once it is initially retrieved, but this caused the app to crash when the record was deleted while there were still living references to it. A better mechanism to deal with database records that do not exist should be implemented, but to fix the issue the database is allowed to emit null to the repository when the record does not exist but the repository only emits non-null records. The schema was also changed to cascade deletions to segment and marker records that were linked to the deleted session record.

- Task 2: This is a left-over task from the last milestone. Users should be able to delete profiles (renamed to templates) so that they can fix mistaken creations. This involved declaring database interface function signatures for template record deletion, then a corresponding function in the template repository. A submenu was created on the Template Page to house the button to trigger the action. See the session deletion task discussion about the crash deleting database records in use causes and its fix. Sessions linked to templates have their connections broken but are not deleted.
- Task 3: Timeline visualization was not completed due to lack of time. Backend UI State and Viewmodel began production for the new feature. Intended to be completed during the next milestone.
- Task 4: Immediate feedback from the evaluators identified higher priority work items. Fixes to address app failures necessitated that this feature be delayed, and before work on it could begin, the client decided that other features to address the evaluator feedback should be prioritized over it. Currently, there are no plans to implement this feature in the remaining time of the semester because the issue that editing the timeline of a completed task was supposed to address was partly mitigated by the ability to delete sessions (so that erroneous sessions will not affect app predictions).
- Task 5: This task was undertaken to implement a page that shows the user their completed tasks as well as information about them. In the previous milestones, completed tasks were displayed on the same page as upcoming and in-progress tasks. This landing page would quickly become cluttered as more tasks were scheduled, executed, and finished, so we moved completed tasks to a separate tab. Additionally, we wanted to add the capability to display the user's task performance over time.
- Task 6: Immediate feedback from the evaluators identified higher priority work items. Fixes to address app failures necessitated that this feature be delayed, and before work on it could begin, the client decided that other features to address the evaluator feedback should be prioritized over it. The viability of custom task profile (renamed to template)

fields in improving app estimate accuracy was speculative and the amount of effort it would take to implement it is high. The client decided that development effort would be better spent on other features, particularly those identified by the evaluation.

- Task 7: Immediate feedback from the evaluators identified higher priority work items. Fixes to address app failures necessitated that this feature be delayed, and before work on it could begin, the client decided that other features to address the evaluator feedback should be prioritized over it. We also determined that the current iteration of the estimation algorithm would need a larger set of user data to help tune the weights and, therefore, decided to collect more data before making any changes.

- Task 8: As part of the software development process, we gathered feedback from users who were willing to test the application. We were able to find six evaluation candidates (unfortunately the pool is greatly reduced because of the Android phone requirement). Of the six, two did not use the app and one created a task but never started it. While this indicates uncommitted evaluators, it also indicates that ClockWork is not easily adopted into a user's lifestyle, which was a concern. The evaluator who never started the task they created, reported that they needed a reminder feature for the app to be usable. One of the more committed evaluators requested an automated session creation feature and that the previous estimate be the default for the next session. Another evaluator reported that the UI elements were too big. Unfortunately none of the evaluators used the app enough to generate helpful data to use in improving and evaluating the app's session duration estimator.

- Task 9: We have created the poster for the senior design showcase and included the following sections: Motivation, Abstract, Tools Used, System Architecture, Future Improvements, and screenshots of the application's user interface. The poster is intended to detail the motivations behind the application and the methods used in its design for any potential audience.

- Task 10: On evaluator phones, the live notification that displays the elapsed time for a running task timer would either 1) stop updating the elapsed time after a while or 2) increment inconsistently and drift very far behind. This behavior could not be reproduced in the dev environment, so it was not possible to receive error reports or log printouts. Several attempts to address this issue failed, including changing timer controls to use Intents, changing the start service strategy, and adding a wakelock. Two new apps were created to test service strategies in isolation. Testing these apps revealed that the incrementation strategy was flawed because it depended on perfect CPU scheduling, but even though one used the same strategy as ClockWork the behavior where the notification stopped updating could not be reproduced. The incrementor was replaced with a thread that computed the difference from the current time to the start then waited at least one hundred milliseconds. In addition, the final refactor changed the

TimerService to use a tagged union to represent its state so that invalid states were unrepresentable and state transitions were atomic. Since then, no drifting or stopping has been observed in the timer notification.

- Task 11: One evaluator uses a Samsung S9 running Android 10, and they could not open the ClockWork without it crashing. We developed ClockWork to adhere to Android API level 26 capabilities so that all Android versions 8 and above would be supported, but we only used Android 16 and 14 during development. The issue was a SQL query that was not supported in Android 10 and under that used window functions. The query was replaced with one that was less performant but was supported by older Android versions.

- Task 12: One evaluator uses an Android System font scale value one increment greater than default and that, in combination with their phone's screen dimensions, caused numerous visual elements to be squished and unreadable. The implementation we were using to construct most of the UI components that contained the problematic elements was not compatible with the fix (dynamically rearranging the elements onto new lines without orphaning grouped elements). Ultimately, a less Jetpack Compose idiomatic implementation was written that added new rows dynamically without compromising the container's visuals. While we were about it, we started addressing text elements that displayed user-entered names of sessions and profiles (renamed to templates) not restricting their contents to their intended size (e.g. sessions with really long names would take up the entire report page).

- Task 13: ClockWork's philosophy was originally to emphasize the distinction between elapsed time working and on break; they would be displayed separately throughout all of the UI and the timer page only displayed the elapsed time spent working. To address confusion between the "On Break" and "Suspended" states, we agreed that it would make the most sense to unify work time and break time and clearly show that the time it took the user to complete the task included the time where they were on break. The timer page was changed to show the total elapsed time as well as how long it has been since the user began the current segment (e.g. how long their current break has been). The started session UI items in the todo list page were changed so that the former position for worktime now shows the total time and the former position for breaktime now shows the percentage to the user estimate. The completed session UI items in the user statistics page were changed so that the former position for worktime now shows the total time and the former position for breaktime now shows the user estimate error as +/-HH:MM.

- Task 14: One evaluator reported that selecting user estimates felt clunky and imprecise. In an attempt to improve how quickly the user can select the values they want, the user estimate selector was changed so that tapping any visible value would instantly select it. This makes performing microadjustments (e.g. one second to the next) very fast, precise, and easy, but it does not entirely fix the clunkiness of selecting an estimate. We have not yet selected a course to take, but one solution might be to add more friction to the “fling” scrolling so that it stops sooner.

- Task 15: A common complaint among the evaluators was that once they completed a session, they could not edit its properties (e.g. name, color, and what template it is linked to). Evaluators would create their first task, complete it, and then realise that they should have made a template for it but could not assign it to the new template. To fix this, we changed the edit session page to support completed tasks and to not enable changes to the user estimate once they started the task.
- Task 16: As per feedback from Dr. Chan and with the intent to make the terms used in the application more clear, each instance of the term “Profile” has been replaced with “Template”. This change more clearly reflects the intended use of Task Profiles as a means for grouping similar tasks together.

Discussion of Contribution:

- Anthony: I helped to create the Design Showcase poster and make updates to our website to include it. I also worked on the UI of the application to address persistent visual issues, such as long task or template names taking up too much screen space on the relevant pages. Moreover, I replaced each instance of the word “Profile” in the application with “Template” to reflect feedback from Dr. Chan and make the terminology more intuitive for users.
- Christian: I renovated the user statistics and history page. I wanted to implement graphical features that would aid the user in tracking their task performance over time, taking inspiration from fitness apps that show the user a graph that tracks their fitness level during their past workouts. To do this, I researched graphical libraries that offer rich features such as line graphs with animations and touch responsiveness since the built-in graphs in Jetpack Compose are rudimentary. I quickly settled on a line graph to display the accuracy scores of completed tasks, and then refactored the user stats and history page and view model to make use of the accuracy data.
- Peter: I recruited, communicated with, and compiled the feedback from the evaluators for this milestone’s evaluation. I also worked to fix the issues the evaluators were facing as soon as possible. My largest undertaking was fixing the timer notification not keeping time. See the Task 10 discussion above for details about my activities in that effort. Although this was not included in the milestone tasks, I also worked to improve the live notification by upgrading its importance to not be relegated to the “Silent” section while still not notifying the user. I finished implementing the session and template deletion, made completed tasks editable, made the user estimate only editable before the session starts, and changed the estimate selector so that users can tap on the value they want to

select, changed to UI to unify break time and work time. I also did a lot of refactoring and implementation quality improvement.

- Pierson: I began development on the task timeline visualization page. I developed the according UI state class and View model for the visualizer. I also offered opinions in regards to changing scope and milestone tasks.

Task Matrix for Next Milestone:

Task	Anthony	Christian	Peter	Pierson
1. Test/demo of the entire system	25%	25%	25%	25%
2. Conduct evaluation and analyze results	25%	0%	75%	0%
3. Create user/developer manual	25%	25%	25%	25%
4. Create demo video	25%	25%	25%	25%
5. Finish completed session timeline visualization	0%	0%	0%	100%
6. Finish user history and statistics page	0%	100%	0%	0%
7. Session reminder notifications	0%	0%	100%	0%
8. Show last duration and estimate error and other helpful information when estimating a new task	0%	0%	100%	0%
9. Finalize UI	100%	0%	0%	0%

Discussion of Next Milestone Tasks:

- Task 1: To complete the development of the application in this milestone, we plan on thoroughly testing each component of the application and how they interact with each other to ensure smooth operation for the end user. We will address any standing issues in the code and demonstrate the functionality of each component.

- Task 2: As part of the final set of evaluations, we plan to select a few more testers to work with the app and gather feedback from their experience to determine which aspects of development to prioritize.

- Task 3: To make adopting the app as easy as possible, a user manual will be created to walk the user through the app features and explain the app philosophy. Additionally, we will compose a developer manual that explains the processes and architecture of ClockWork to bring the next developer team up to speed.

- Task 4: We intend to create a video production to demonstrate the application and its features. This video will feature an extended and edited overview of the app's goals and features.

- Task 5: We intend to complete the task timeline visualization for users to view from the history details page of a task or by selecting the details button from the task completion page. The frontend UI still needs to be designed and the methods for querying the database for our timeline markers still needs to be implemented.

- Task 6: We intend to implement the ability for users to filter completed tasks by profile within the user history and statistics page. Doing so will require the page to be refactored to allow for profile selection, and to update the page with the filtered data.

- Task 7: Two of the evaluators reported that the app would not be usable to them if it did not remind them of when they wanted to start their task sessions. The intention is to have added scheduled reminder notifications early in the milestone (by the end of the first week) so that these evaluators can use the app more in their routines and generate more in depth feedback.

- Task 8: To help the user make more thoughtful estimations, the estimate selector in the new task form should show the user some information about their historical performance with tasks from the current template. The selector should also default to the previous session (from the same template)'s estimate so that the user can iterate easier.

- Task 9: As part of the polishing process, we intend to take a complete look over the application's UI and make sure that there are no remaining visual bugs or inconsistencies, as well as make any necessary changes based on feedback from the evaluations, such as reducing the size of UI elements.

Meeting Dates with Client During Current Milestone:

- see Faculty Advisor Meeting Dates below

Client Feedback on Each Task of Current Milestone:

- see Faculty Advisor Feedback below

Meeting Dates with Faculty Advisor During Current Milestone:

- Meeting 1: Oct 15, 2025
- Meeting 2: Oct 24, 2025

Faculty Advisor Feedback on Each Task of Current Milestone:

- Task 1:
- Task 2:
- Task 3:
- Task 4:
- Task 5:
- Task 6:
- Task 7:
- Task 8:
- Task 9:
- Task 10:
- Task 11:
- Task 12:
- Task 13:
- Task 14:
- Task 15:
- Task 16:

Faculty Advisor Signature: ________________________________ Date: ________

Evaluation by Faculty Advisor

- Faculty Advisor: detach and return this page to Dr. Chan (HC 209) or email the scores to pkc@cs.fit.edu
- Score (0-10) for each member: circle a score (or circle two adjacent scores for .25 or write down a real number between 0 and 10)

Anthony	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Christian	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Peter	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Pierson	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10

Faculty Advisor Signature: ________________________________ Date: ________