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. Design Task Session List UI	60%	0%	0%	60%	40%	Completed session list (postponed until profile implementation)
2. Implement Task Session List UI in Swift	0%	0%	0%	0%	0%	
3. Implement Task Session List UI in Kotlin	60%	0%	0%	0%	100%	Completed session list (postponed until profile implementation)
4. Design Task Session Timer UI	90%	0%	0%	80%	20%	Task information page to be elaborated

5. Implement Task Session Timer UI in Swift	0%	0%	0%	0%	0%	
6. Implement Task Session Timer UI in Kotlin	60%	0%	0%	80%	20%	Initial info display
7. Design (Initial) Task Session Completion UI	100%	0%	0%	80%	20%	
8. Implement (Initial) Task Session Completion UI in Swift	0%	0%	0%	0%	0%	
9. Implement (Initial) Task Session Completion UI in Kotlin	100%	0%	100%	0%	0%	Save timer, %over-estimate, %improv
10. Implement Session Data Serialization and Persistence in Kotlin	0%	0%	0%	0%	0%	Next Milestone
11. Implement Session Timer in Kotlin	60%	0%	0%	100%	0%	Add suspend and mark functionality
12. Explore how the app interprets user data to make estimations and tracks progress	80%	100%	0%	0%	0%	Test and refine model
13. Explore how the app treats user estimations	80%	100%	0%	0%	0%	Test and refine model

Discussion of Current Milestone Tasks:

Task 1: In its current form, all unfinished task sessions are displayed in a single list and finished task sessions are not represented. The former is because task profiles have not been implemented and will not be implemented until individual task sessions are more fleshed out. The latter is because task completion logic, and most task model logic, is not implemented yet.

Task 2: The iOS UI for the Task Session List page was intended to be complete, however after a reconsideration of how much time and effort will be put into improving the visuals of the UI overtime, iOS development will be held for a later date.

Task 3: The Android UI for the Task Session List page is now implemented in Kotlin as it was designed in the mockup.

Task 4: Mockups depicting the task timer page during various states of the timer were created. The task info element of the page while the timer is in its “INIT” stage was left undefined because logical task representation has not yet been elaborated in enough detail to populate the fields.

Task 5: The iOS UI for the Task Session Timer page was intended to be implemented. See discussion for Task 2.

Task 6: The Android UI for the Task Session Timer page is implemented in Kotlin including all elements depicted in the mockups except for the task info panel which will be implemented when task definition is elaborated.

Task 7: Models depicting the Task Session Completion page will be created so that the design of this part of the UI will be unified between the iOS and Android apps. Elements of this page concern how the app provides feedback to the user which is a question that is being explored during this milestone. Therefore, those elements will not be present unless progress that effort proceeds farther than anticipated.

Task 8: See Task 2.

Task 9: Following on from the previous milestone, a prototype user interface was implemented in Kotlin. I used the available design document as a blueprint for the Task Session Completion screen. Fully implementing the backend logic to properly save and display the user’s ending time, their initial estimation, %overestimation, %improvement from recent averages, as well as button functionality, goes beyond the scope of this milestone. These features will evolve as the application evolves, and expect to include them in the next milestone.

Task 10: This task proved to be out of scope for the current milestone and will be addressed in the next one.

Task 11: The backend components for the Android app that provide the task session timer functionality is implemented in Kotlin. This version of the session timer only includes the ability to start, pause, unpause, and stop. Regardless, considerations have to be made to support timeline events in future iterations.

Task 12: One of the major goals of the system is to make estimates for hypothetical tasks. The first step is to investigate and determine which methods will be employed to achieve this. After some initial research, we have determined the most promising approach to be an exponential smoothing algorithm that takes a weighted average of past estimates and actual user task completion times to give an initial estimate for the current task. The algorithm follows the formula: $T_{new} = [\alpha T_{prev\ time} + (1 - \alpha)T_{prev\ estimate}] * \frac{difficulty}{3}$. Alpha values ($0 < \alpha < 1$) greater than 0.5 put more weight on recent data to account for the user improving their performance over time. The difficulty multiplier modifies the estimated time to be higher/lower based on how hard/easy the task is perceived to be. This is an initial model and will likely be modified as real data is collected and analyzed

Task 13: One of the major goals of the system is to give feedback to the user about their time estimations. We must determine the app's philosophy regarding user accuracy such as in what manner it provides feedback, what are the thresholds for good or bad estimations, how we gauge user improvement, etc. So far, this is something to continue analyzing as we implement and test the model established in the previous task.

Discussion of Contribution:

Anthony: I spent this milestone working on prototyping the algorithm the application will use to estimate the time it should take a user to complete a certain task. As described in Task 12, the current formula is: $T_{new} = [\alpha T_{prev\ time} + (1 - \alpha)T_{prev\ estimate}] * \frac{difficulty}{3}$. This accounts for the user improving their task completion times over time and the difficulty of a specific task on a scale from 1 to 5. This model will be refined based on collected data as the project develops.

Christian: I coded the initial prototype of the *Task Session Completion Screen*, focusing on building a scaffold that accurately reflects the UI design diagram. The screen – screens are called *pages* in Compose – has not implemented functionality, the time at which the user presses the *finish* button is not displayed, and the *View Details* and *Continue* buttons don't work. Future iterations will include the requisite logic.

Peter: I worked with Pierson to design all of the UI pages assigned for the current milestone and created visual mockups. I established the initial code organization and implemented the initial timer functionality, timer UI, and connected the two.

Pierson: Worked with Peter to refine and understand the designs he made. Improved upon my initial Kotlin task list to have it confine closer to the provided design. Assisted with development of the timer page. Provided merging work for all GitHub branches.

Task Matrix for Next Milestone:

Task	Anthony	Christian	Peter	Pierson
1. Cohesion between and functionality of session, timer, and completion pages	0%	20%	0%	80%
2. Implement Session Data Serialization and Persistence in Kotlin	0%	0%	100%	0%
3. Implement New Task Session UI in Kotlin	0%	100%	0%	0%
4. Implement user progression evaluation in Kotlin	60%	0%	0%	40%
5. Implement task time estimation in Kotlin	60%	0%	0%	40%

Discussion of Next Milestone Tasks:

Task 1: Implement navigation between the Task Session List, Task Timer, and Task Completion pages such that interacting with an entry in the task session list will take the user to the proper timer page, completing a task in the timer page will take the user to the task completion page and populate its fields with the accurate information regarding the session, and enabling navigation out of the task timer and completion pages back to the task list. Events in the timer must also reflect in the list page: the times must update, and the entry must be removed when the task is completed.

Task 2: The backend components for the Android app that handles saving task session data such as timer progress, the session name, the session color, etc will be implemented using Kotlin. This will involve an investigation into persistent local data storage solutions as well as an elaboration of the logical representation of tasks.

Task 3: The Android UI for the New Task Session page will be implemented using Kotlin. This will enable the user to specify the fields of task instances. At the moment, tasks are distinct units as opposed to instances of a task profile which link a collection of similar tasks together. As such, a lot of the task properties will be repeated between multiple tasks. This page will be accessible from the task list page as well as the task timer page before the timer has been started. Entering the page from the task list page will instantiate an anonymous task, however, which will be finalized into the serialized task when the fields have been filled out.

Task 4: Implement the components for determining user estimation progress based on the models developed in the current milestone. User estimation progress will reflect user performance among all task sessions and will indicate if a user is improving their ability to estimate the time they take to complete tasks. This might involve implementing an additional UI page.

Task 5: Implement the components for estimating task duration based on the models developed in the current milestone. Algorithmic time cost estimation will incorporate data from all task sessions until task profiles are implemented.

Meeting Dates with Client During Current Milestone:

- Meeting 1: Mar 7, 2025
- Meeting 2: Mar 24, 2025

Client Feedback on Each Task of Current Milestone:

- see Faculty Advisor Feedback below

Meeting Dates with Faculty Advisor During Current Milestone:

- Meeting 1: Mar 7, 2025
- Meeting 2: Mar 24, 2025

Faculty Advisor Feedback on Each Task of Current Milestone:

- Task 1: we discussed – no feedback here. I am fine with this
- Task 2: we discussed – I'm glad to see you making these sorts of decisions for when to move forward and when to delay
- Task 3: Looks good
- Task 4: I like the mockups you showed me. I have no additional feedback – you're capturing the essence of what I was looking for here.
- Task 5: see feedback on Task 2
- Task 6: this is fine
- Task 7: this is fine
- Task 8: see task 2
- Task 9: see feedback on Task 4. No additional feedback at the moment
- Task 10: this is fine
- Task 11: no additional feedback here
- Task 12: we'll want to think more about the formula, but I think this is a good start! Will review the paper Dr. Chew sent me and see if it has any additional info that could be

useful, though I think it is more about perception of time than estimating time. I'll share with you as well. Time estimation is tricky, but if your design is robust enough, changing out the formula should be pretty straightforward. I don't see this (formula) being coupled to anything else in a way that is deleterious.

- Task 13: I look forward to hearing more about the philosophy regarding user feedback. It may be that the Time Perception paper Dr. Chew sent me (which is really just a summary of other research) could be helpful here as well

Faculty Advisor Signature: ______________________________ Date: ________