# ClockWork Time Estimation

Anthony Menendez Christian Ott Peter Stelzer Pierson Hendricks

Advisor: Dr. David Luginbuhl

## Task: UI Design

#### **Design UI for:**

- Task Session Timer Page
- Task Session List Page
- Task Session Complete Page

# Unify interface between Android and iOS apps

#### **Task Session Timer Page**



#### Task Session List Page

	1	
Fades in and out	TASK SESSIONS	$\square \oplus$
	STARTED	
	Task Name	
	00:51 + 00:10	
	Task Name SUSPENDED 00:51 + <u></u>	
	UPCOMING	
	Task Name	
	3:15 3:45	F
	Task Name	
	■ TODAY 3:00PM 3:15 ■ 3:45	F
	Task Name	
	3:15 S:45	F
	Task Name	
	1/2/3456 2:00pm 3:15 3:45	<u>J</u> EJ

#### Task Session Complete Page



## **Task: UI Implementation**

#### Implement UI for:

- Task Session Timer Page
- Task Session List Page
- Task Session Complete Page

#### In the Android app.

# Task SessionTimerPage,<br/>prototype



Task Session List Page, prototype



#### Task Session Complete Page, prototype

9:48 🛇 🗳 🛇 🗂 🔹

Task Name Completed! 741

#### 00:00

You estimated 00:00

--% overestimate

--% improvement from recent averages

View Details

Continue

# **Task: Implement Timer Logic**

- Increment minutes & hours
- Start
- Stop
- Pause

#### **Video Demonstration**

#### **Task: Algorithmic Time Estimation**

• Exponential smoothing formula for estimating the time it will take a user to complete a task:

• 
$$T_{new} = [\alpha T_{prev_{estimate}} + (1 - \alpha) T_{last_{time}}] * (difficulty / 3)$$

- Prioritizes more recent data
- Difficulty acts as a multiplier for estimated time
- Refined through testing + user data

#### **Task: Explore User Progress Evaluation**

- Compare user estimates to actual times
- Within profile, user will be notified of consistent patterns in the differences between past estimates and actual times for a task
  - Ex: If user often underestimates certain task by 30 minutes, this will be reflected within the user profile
- Account for outliers (forgot to turn timer off, variation in tasks)

#### M2 Task Matrix (1)

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%	

#### M2 Task Matrix (2)

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

#### M2 Task Matrix (3)

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

# Milestone 3 (Apr...)

- Cohesion between and functionality of session, timer, and completion pages
- Implement Session Data Serialization and Persistence in Kotlin
- Implement New Task Session UI in Kotlin
- Implement user progression evaluation in Kotlin
- Implement task time estimation in Kotlin

#### Milestone 3

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%