

# Clock Work Time Estimation

Anthony Menendez  
Christian Ott  
Peter Stelzer  
Pierson Hendricks

Advisor: Dr. David Luginbuhl

# Task: System Cohesion

- Proper State Management
- Timer Data Modification
- Completion Data

# Task: Data Persistence

- Android Room Database
- New Representation for Task Structures
  - Old structure did not support anticipated features and was inelegant
  - New structure has to be serializable

# New Task Structure

## Task Composition

### Task Properties

~id~  
name  
difficulty  
due date  
color  
user estimation  
app estimation

### List of Segments

type  
start time  
duration

type  
start time  
duration

Sat 3/3/2015

00:00 ET  
9:31am ● Task Begin

00:24 ET  
9:55am ● Marker 1

00:53 ET  
10:24am ○ Break

00:53 ET  
10:28am ○ Resumed

01:43 ET  
11:18am ○ Suspended

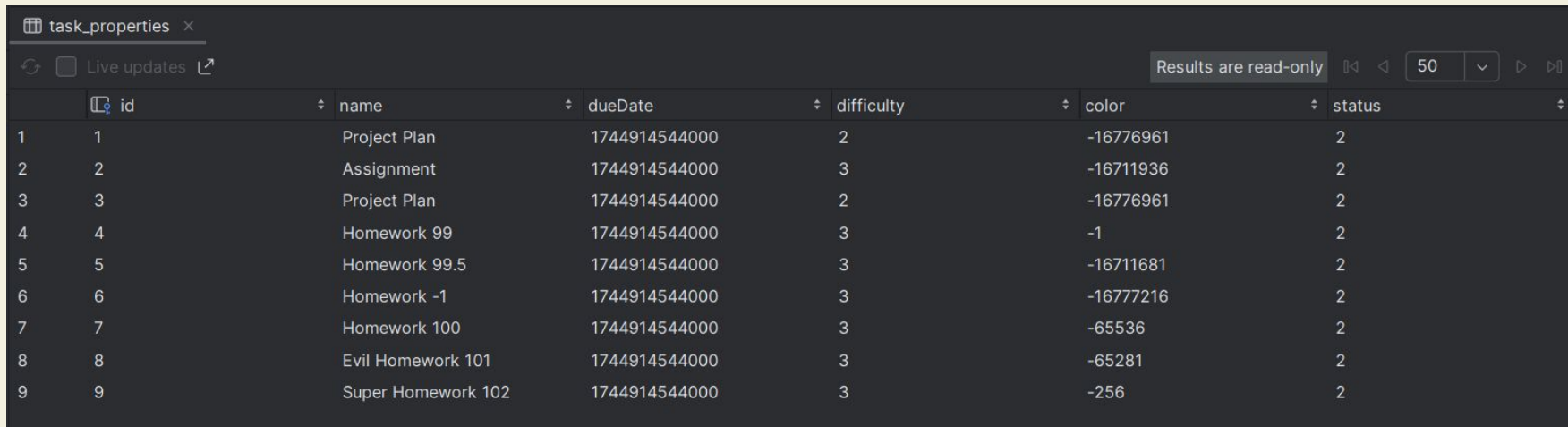
Sun 3/4/2015

01:43 ET  
2:22pm ○ Resumed

02:01 ET  
2:43pm ● Task Complete

# Task: Data Persistence

- Android Room Database
  - Serialized Structures



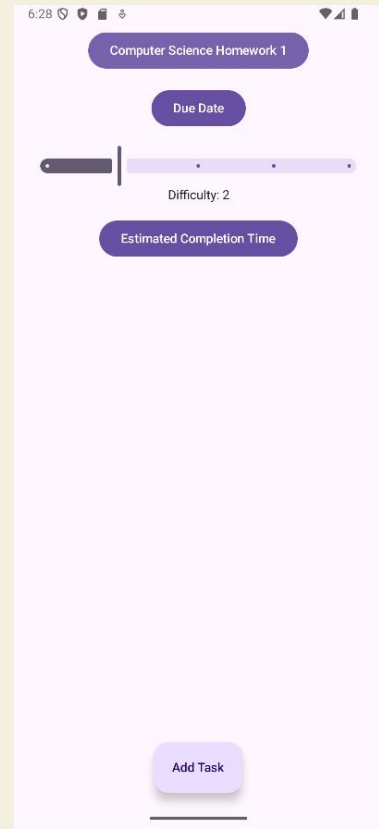
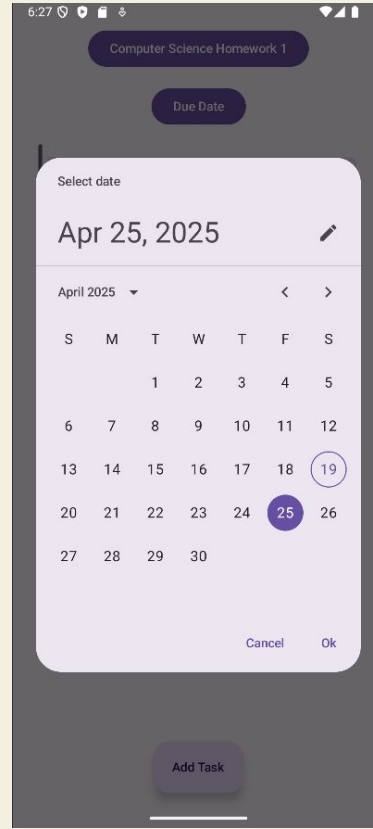
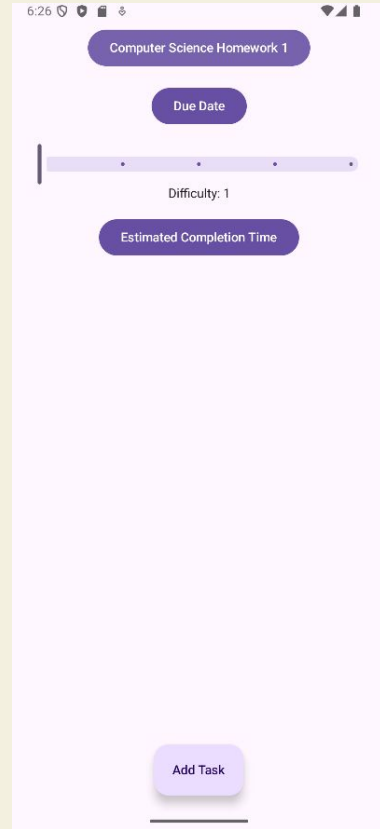
The screenshot shows the 'task\_properties' table in an Android Studio interface. The table has 9 rows and 7 columns: id, name, dueDate, difficulty, color, and status. The interface includes a 'Live updates' toggle and a 'Results are read-only' status bar. The table data is as follows:

	id	name	dueDate	difficulty	color	status
1	1	Project Plan	1744914544000	2	-16776961	2
2	2	Assignment	1744914544000	3	-16711936	2
3	3	Project Plan	1744914544000	2	-16776961	2
4	4	Homework 99	1744914544000	3	-1	2
5	5	Homework 99.5	1744914544000	3	-16711681	2
6	6	Homework -1	1744914544000	3	-16777216	2
7	7	Homework 100	1744914544000	3	-65536	2
8	8	Evil Homework 101	1744914544000	3	-65281	2
9	9	Super Homework 102	1744914544000	3	-256	2

## **Video Demonstration**

# New Task Session Page, Prototype

*Scrolling wheel in Kotlin example: for user Estimated Completion Time. (not implemented in the Page):*



# Task: Implement Time Estimation

- Estimation logic formula created as a Kotlin component
- Plan on merging with corresponding UI element
- Possible supplemental formulas to improve accuracy
  - Isolate assignments based on difficulty
  - Swap out previous estimate in formula with more data



# M3 Task Matrix

Task	Completion %	Anthony	Christian	Peter	Pierson	To do
1. Cohesion between and functionality of session, timer, and completion pages	70%	0%	0%	0%	100%	Await further development
2. Implement Session Data Serialization and Persistence in Kotlin	100%	0%	0%	100%	0%	Refactor and integrate with app functions
3. Implement New Task Session UI in Kotlin	70%	0%	100%	0%	0%	Time estimate interface
4. Implement user progression evaluation in Kotlin	0%	0%	0%	0%	0%	
5. Implement task time estimation in Kotlin	30%	100%	0%	0%	0%	Implement into UI

# Milestone 4 (Apr...)

- Integrate data storage and structures
- Implement task editing and complete creation
- Implement timer segments
- Add timer estimation
- Link menus and components properly

# Milestone 4 Matrix

Task	Anthony	Christian	Peter	Pierson
1. Cohesion between and functionality of session, timer, and completion pages	0%	0%	0%	100%
2. Integrate data storage with task creation, editing, and timer execution	0%	0%	100%	0%
3. Finish New Task Page and Create Edit Task Properties Page	0%	100%	0%	0%
4. Timer execution segments integration	0%	0%	100%	0%
5. Integrate task time estimation into task creation and the corresponding UI	100%	0%	0%	0%

