

Tech Assignment - iOS

Overview

Tech Ticks is an engaging and intuitive task management app designed for iOS. The application leverages the powerful Core Data framework for efficient data management and provides a user-friendly interface to help users efficiently manage their tasks. It incorporates local notifications to alert users about task deadlines, ensuring they stay on top of their schedules.

Key Components

Home Screen Features:

1. Dynamic Task List Display:

- Tasks are retrieved from Core Data and displayed in a UICollectionView.
- The list offers a filter to view tasks based on their status: Upcoming, Ongoing, and Completed.

2. Task Details:

- Each task displays the following:
 - Title
 - Description
 - Start and End Date and Time

3. Task Status Management:

- Task status updates automatically based on the current date and time.
- Users can mark only 'Ongoing' tasks as completed.

4. Interactive Task Options:

- Long press on a task allows users to Edit or Delete it.
- Only 'Upcoming' tasks are editable.

5. Add Task CTA (Call to Action):

- A prominent button for adding new tasks.

Add or Edit Task Screen:

1. User Interface Components:

- Text fields for entering task title and description.
- Date picker for selecting start and end dates.
- Options to set the task status.

2. Core Data Integration:

- All tasks are stored and managed in Core Data.

3. Local Notifications:

- The app schedules local notifications to alert users at the start time of each task.

Enhancements /Technical Requirements

1. Core Data Usage:

- Efficiently utilise Core Data for CRUD (Create, Read, Update, Delete) operations on tasks.

2. UI/UX Design:

- Ensure a clean, intuitive design for the CollectionView layout.
- Implement an interactive and responsive interface for task management.

3. Local Notifications:

- Implement local notifications that are triggered at the designated start times of tasks.

4. Error Handling:

- Robust error handling to manage any discrepancies or issues during data retrieval or manipulation.

5. Code Quality:

- Write clean, modular, and well-documented code.

- Follow best practices in iOS development.

Deliverables and Expectations

1. Source Code:

- Well-structured and commented source code.

2. Documentation:

- Detailed documentation covering the app's architecture, Core Data model, and any specific libraries used.

3. Testing:

- Unit and UI tests to ensure the application's functionality.

Evaluation Criteria:

- **Functionality:** How well does the application meet the outlined requirements?
- **UI/UX Design:** Is the application intuitive and visually appealing?
- **Code Quality:** Clarity, structure, and use of best practices in coding.
- **Error Handling and Robustness:** How well does the app handle unexpected scenarios?
- **Innovativeness:** Creative solutions to common problems in task management apps.

General Development Tips

1. Stay Updated with Swift and iOS Development:

- Regularly visit Apple Developer Documentation and the Swift Blog.

2. UI/UX Design:

- Refer to Apple Human Interface Guidelines.

3. Effective Use of Core Data:

- Check out Ray Wenderlich's Core Data Tutorial.

4. Writing Clean Code:

- Follow Swift API Design Guidelines.

5. Testing Your Application:

- Learn from Apple's Testing with Xcode.

6. Version Control with Git:

- Explore GitHub's Learning Lab.

7. Stay Current with iOS Updates:

- Subscribe to [iOS Dev Weekly](#).

8. Join Developer Communities:

- Participate in forums like [Stack Overflow](#) and Reddit's iOS Programming subreddit.

9. Performance Optimisation:

- Use Instruments in Xcode.

10. Accessibility:

- Consult Apple's Accessibility Developer Resources.

As you embark on this project, remember it's more than just a task management system—it's a canvas for your innovation. Let your coding skills and creativity intertwine to create a user-friendly, effective solution. This is your chance to shine and make an impact.

Good luck!