CS 329E
Semester Project

Quizlet
StudyCard

For your semester project, you must design and develop an iOS application. While the project has a minimum set of required features, you are welcome to focus on the areas of app development that are most relevant to your personal interests. The application you build must be worthy of the amount of time available to design and build it.

You will be working on a team of 4 students. By now, you should have already formed your groups.

It is very likely that you will want to start designing and coding your app before we have discussed some of the frameworks in class. You are responsible for educating yourselves on how to get started with those frameworks. Use the Internet, or ask for tips from the TAs.

Basic application requirements:

- You cannot enhance an existing product or something that you have already written. Your code for this project must be original.
- Your app must include a login/register path (with email address and password) using Firebase.
- Your design must include a "Settings" screen. Define at least three behaviors in your app that the user can modify through this screen, such as "dark mode", font style, color scheme, sound on/off, etc. The app should have default settings for these. (This is in addition to any settings you include in a user profile, if you choose to incorporate that.)
- Your app must have a well-designed, professional-looking user interface that includes non-default fonts, colors, and styles integrated attractively and consistently across the project. (It is okay to use default fonts and colors for basic controls, such as the nav bar or settings, but you will be penalized if the majority of your app uses defaults.)
- Your app must be functionally "complete". If you create something (a user profile, a post in a feed, a friend in a contacts list, etc.), you should also include a method for editing or deleting it, provided it makes sense for the user to want to do that.
- The code for your app must meet quality expectations. You will be penalized if your app crashes while being graded, if there is missing or incomplete functionality, and if you do not conform to the Swift Coding Standard.

Technical Requirements:

Major Elements: Your app must include *at least two* of the following.

- Use of Core Data to persist a significant amount of data across launches - more than just user configuration or settings data. Flashcards

- Implementation of a user profile path, including the ability for the user to create a profile, take a picture using the camera or select a preexisting picture from the photo library, and edit a profile.
- Use of multithreading in a meaningful way.
- Use of SwiftUI to design the user interface.

Minor Elements:

- You will naturally use labels, buttons, images, and text fields in your app. In addition, your app must also include *at least two* instances of other view types (such as text views, sliders, segmented controllers, date or color pickers, steppers, switches, search fields, bars and bar buttons, etc.)
- Your app must incorporate *at least one* of the following: Table View, Collection View, Tab VC, Page VC.
- Your app must use *at least two* of the following: Alerts, Popovers, Stack Views, Scroll Views, Haptics, User Defaults.
- Your app must include *at least one* of the following frameworks *per team member*:
  - Local notifications
  - Core Graphics
  - Gesture Recognition
  - Animation
  - Calendar
  - Core Motion
  - Core Location / MapKit
  - Core Audio
  - Others (such as QR code, Koloda, etc.) with approval from the instructor

Turning in your project:

Important Due Dates:

- Plan for your team to meet with your TA to present your project for feedback the week of the intermediate checkpoints (Design, Alpha, and Beta).
- Final code will be due by 11:59 pm on the last day of classes.
- Presentations will be made during class throughout the last week (or so) of classes. Presentation order will be determined randomly.
- All final deliverables should be placed together in one directory/folder, zipped up, and submitted on Canvas before the due date/time.

Deliverable 1: Presentation

The last week or so of classes will consist of presentation days, during which teams / individuals will introduce themselves, and then present their projects to the rest of the class. Presentations will be about 10-15 minutes long, and consist of a demo/walkthrough of the app.

- Walk through all implemented paths. Your program should not crash.
- All required functions should be present. Stubs are okay for function dropped or stretch goals not implemented, but explain them. They should look like they would if you planned to continue working on it.
- Teams are strongly encouraged to practice their presentations beforehand. Also, start up Xcode and build your project on any laptop you plan to use BEFORE the start of your presentation so we don't have to wait for you to set it up.

Deliverable 2: Your Code

Deliverable 3: README file

Turn in a README file in your project folder containing:

- Name of project and list of team members
- A list of dependencies (Swift version? Xcode version? Any pods or frameworks to install first?)
- Any special instructions your instructor will need to know in order to test your app
- A "checklist" showing how you implemented each required feature.
- A work distribution table. Each row should contain one major feature of your project. The columns of the table are:
  - Short description of feature
  - Who worked on what percentage of it

Grading of the Project:

- Use the same mechanism to turn this in as you did for your homework assignments: ONE person on your team should attach your team's code to the "Final" assignment in Canvas. **Late work will not be accepted**, so plan for contingency!

Your final release will be graded on the quality of the implementation, both in the code and the user interface. I will be looking for clean, well-structured code, as well as a reasonably well-defined and designed UI. Every team member is expected to attend and actively participate in the presentation (with cameras turned on if done over Zoom). Members who do not participate will be penalized on their final project grade.

Sample README file:

Name of project: Dragon War Game
Team members: Cersei Lannister, Jon Snow, Danaerys Targaryen
Dependencies: Xcode 10.2, Swift 4

Special Instructions:

- Use an iPhone 8+ Simulator
- Before running the app, run "pod install" inside the DragonWar folder where the podfile is located
- Use this test account for logging in:
  email: DragonWarTest@gmail.com
  password: DragonWar
- To test the connection between two players, you need to set two player mode on, and you need to run it on an iPhone and a simulator at the same time (or 2 simulators)

Required feature checklist

- ☐ Login/register path with Firebase.
- ☐ "Settings" screen. The three behaviors we implemented are:
  (fill in) _____
- ☐ Non-default fonts and colors used

Two major elements used:

- ☐ Core Data
- ☐ User Profile path using camera and photo library
- ☐ Multithreading
- ☐ SwiftUI

Minor Elements used

- ☐ Two additional view types such as sliders, segmented controllers, etc. The two we implemented are: (fill in) _____

One of the following:

- ☐ Table View
- ☐ Collection View
- ☐ Tab VC
- ☐ Page VC

Two of the following:

- ☐ Alerts
- ☐ Popovers
- ☐ Stack Views
- ☐ Scroll Views
- ☐ Haptics
- ☐ User Defaults

At least one of the following per team member:

- ☐ Local notifications
- ☐ Core Graphics
- ☐ Gesture Recognition
- ☐ Animation
- ☐ Calendar
- ☐ Core Motion
- ☐ Core Location / MapKit
- ☐ Core Audio
- ☐ Others (such as QR code, Koloda, etc.) with approval from the instructor – list them

Work Distribution Table

Required Feature	**Description**	**Who / Percentage worked on**
Login / Register	Allows user to create account and login	Cersei (100%)
UI Design	Team shared equally in designing the wireframe for the app, and selecting colors and fonts.	Cersei (25%) Dani (25%) Jon (25%) Tyrion (25%)
Settings	Allow user to set one/two player mode, saving lib, add photo to account, logout and delete account	Cersei (50%) Dani (25%) Jon (25%)
Profile path	Allow user to create, edit, and delete a user profile. Includes code for using camera and photo library.	Jon (50%) Tyrion (50%)
Draw	Allow users to draw pictures	Dani (100%)
Instructions	Instructions on how to play the game	Cersei (100%)
One player mode	Allows one player to create a dragon and play against AI	Cersei (80%) Tyrion (10%)

		Jon (5%) Dani (5%)
Connect two players	Allow 2 players to setup a network connection	Dani (50%) Tyrion (50%)
Two player create new scenario	Implement most UI screens, and all navigation, logic, server integration for 2 players interacting to create a scenario together	Dani (90%) Cersei (10%) (Setup for UI/ container view, Eureka integration)
	. . .	