







Pinned! Mobile App Design

Cody Mattern, Anna Shea
Kyle Giacomini, Tyler Albee

Tools Used

- Trello 
 - Project Tracker
 - Was mainly used to visualize and grant a universal understanding of tasks to be done and in what order.
- Visual Studio/Xamarin Studio 
 - IDE
 - Served in development, testing GUI and front end integration(via emulation), and deployment.
 - Complexity of software did cause various issues.
 - Xamarin integration in VS 15 caused slow downs due to xamarin only partially working (VS 17 launched in the middle of the semester).
- OSX 
 - Required for IOS emulation
 - Only problem this caused was financial barrier to entry, otherwise we had better luck building for IOS than android.
- Github 
 - VCS
 - did not cause any major problems throughout project.
- MySQL 
 - Database
 - Much like github, it worked as intended.
- Doxygen 
 - Auto-documentation utility

Methodologies

- Agile ★★★★★
 - Focused on building and integrating single functionality at a time.
- Pair Programmed when possible ★★★★★
 - Due to the widely varied experience throughout the group more experienced group members aided the less experienced members with programming were possible.
 - Found ourselves with few opportunities to do this due to busy schedules all around.

Challenges (iOS Frontend)

- Xamarin documentation was confusing towards cross platform projects.
- Struggled to get Native Storyboard to run initially.
- When confronted with Exception errors, within iOS simulator. The editor lacked information on where the error occurred.
- While Xamarin is doing a great job giving people the ability to work on iOS project through their windows desktop, the mac connector was buggy.
- Overall we lost a lot of time and even code, to Xamarin/Visual Studio issues. We had to rebuild the project a 4 or 5 times. These constant setback hindered our ability to have a polished project.
- Tyler and Anna used xcode to build their storyboards. Issues with importing them into xamarin.

Challenges (Android Frontend)

- Faulty dependencies / failing to acquire through IDE
 - After no solution in IDE could be found ended up reacquired missing build dependencies manually.
- Visual Studio 2015 integration was faulty and only partially working (2017 Launched recently)
 - This caused major slow downs as android frontend had to be written manually.
 - We did end up adopting VS 2017 which worked as intended (notably the designer mode), however we had lost a lot of time to 2015.
- Integration of Google Maps API was tricky
 - Resolved via study of how Google API use and registration works.

Challenges (Backend)

- Learning Node.JS. Should of used python instead.
- Understanding AWS security groups was a difficult task. Couldn't access database for a while.
- Lambda lacks a lot of functionality in terms of other AWS services.
- AWS CloudWatch(logs) is very difficult to navigate
- Since we were using the free tier within AWS, Lambda functions constantly timed out.
 - This caused issues on app, that still havn't been fixed.

Demo functionality

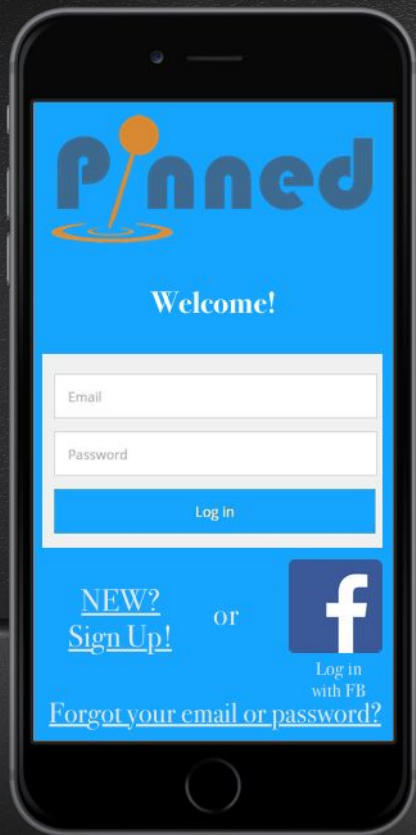
- Working login
 - Login successfully can query database and verify login credentials
- Working User Creation
 - Account creation page and user confirmation(confirmation code sent to email)
- Working map Display
 - Map is being generated via MapKit(iOS) and Google Maps(Android)
 - View is rendered based on location on phone/emulator
- Working pin dropping
 - Long Press Gesture captures event
 - Opens simple pin details page to create Pin

Xamarin iOS Storyboarding: LaunchScreen.Storyboard



Upon loading, the center design will rotate and “unlock”, revealing either the Login Page (if `isAuthenticated` is false) or the Home Page (if `isAuthenticated` is true).

Main.storyboard: isAuthenticated flag set to false



Main.storyboard: isAuthenticated flag is true

