MICHIGAN STATE UNIVERSITY

Project Plan

Mobile GSAP and QC Audit Center v2.0

The Capstone Experience

Team United Airlines Airport Operations

Hani Habhab
Jacob Macbrien
Austin Evans
Tu Le

Department of Computer Science and Engineering
Michigan State University

Fall 2020

From Students…
…to Professionals

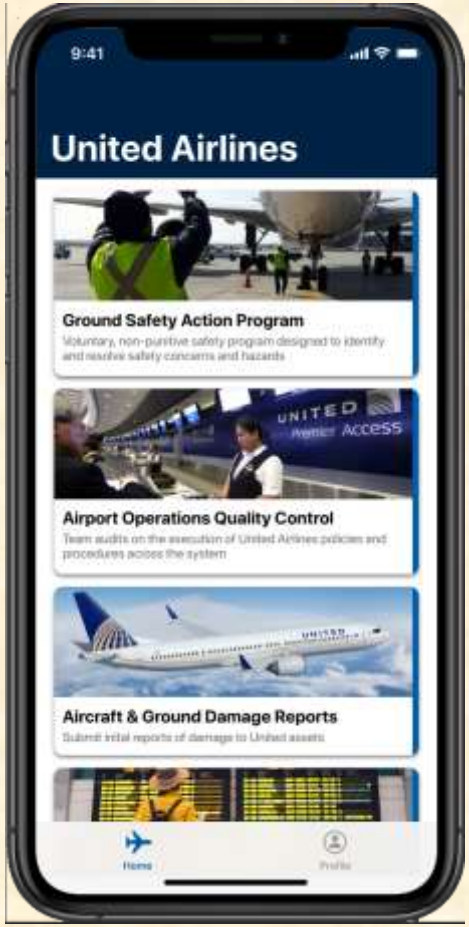
Functional Specifications

- Version2.0 of the Mobile application will allow aircraft ground safety employees to start and audit or report real-time while inspecting the aircraft or any asset.
- Reduce time wasted handwriting reports and transferring them to web-based desktop version.
- Addition of three more reporting and auditing features
- Allow ability to track themselves and their required audits goals every month.

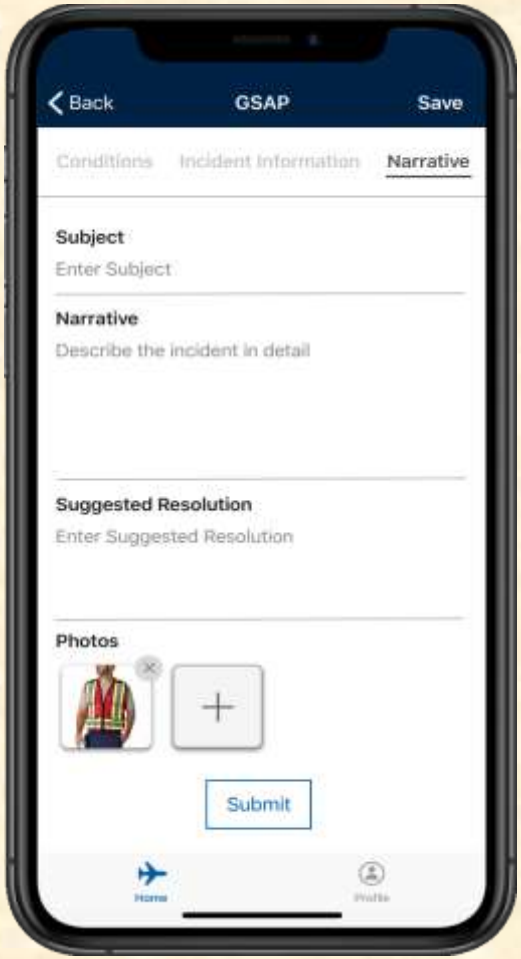
Design Specifications

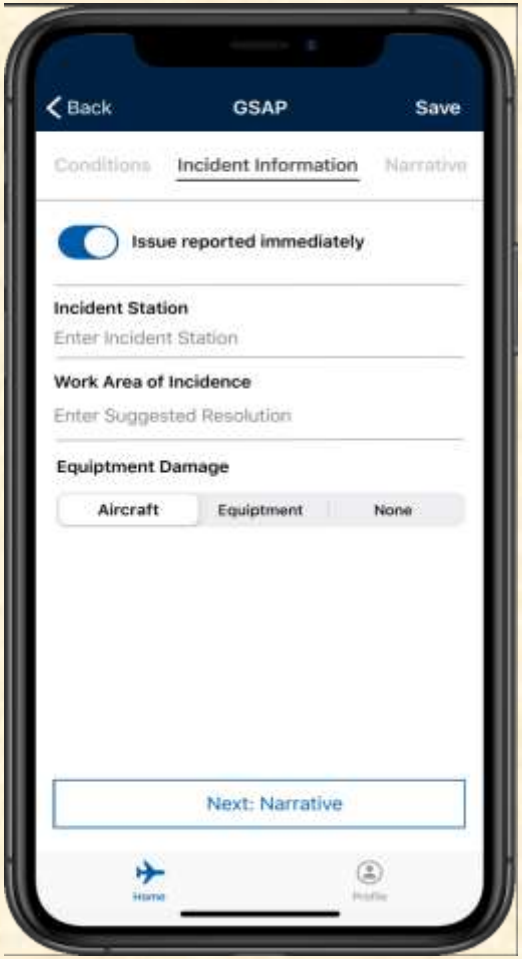
- Redesign of the User Interface to be more user friendly, and ease of use through-out reporting process.
- Expansion of three more features- MOCHA, LOSA, ERS. This will allow for more flexibility on larger allowance for more reports to get done
- Allow as much information to be entered and processed
  - Pictures or videos
  - Comment Sections
  - Auto-fill dates
- Creation of dashboard to allow the user to check how many audits/reporting they have completed and meet their monthly goals.
- Ability to save drafts of forms that are currently in progress.

Screen Mockup: iOS & Android form selection page

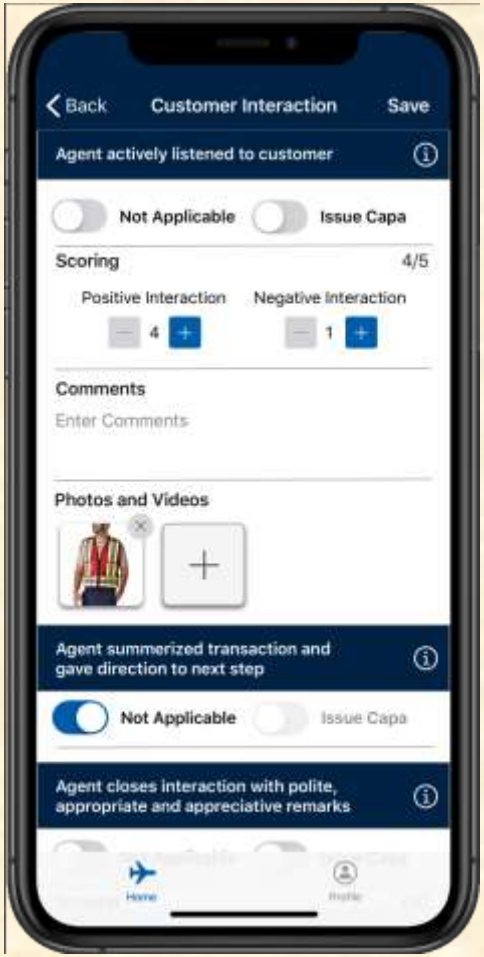


Screen Mockup: iOS GSAP Form

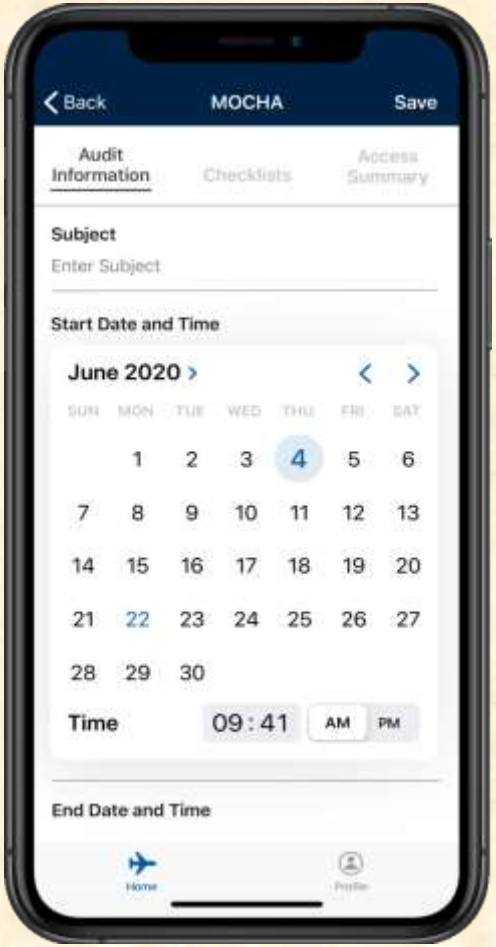




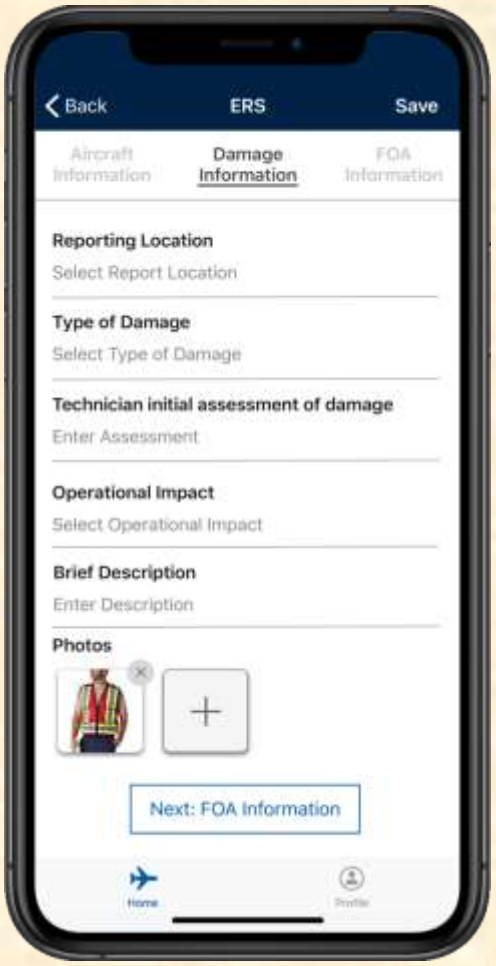
Screen Mockup: iOS QC Audit Form

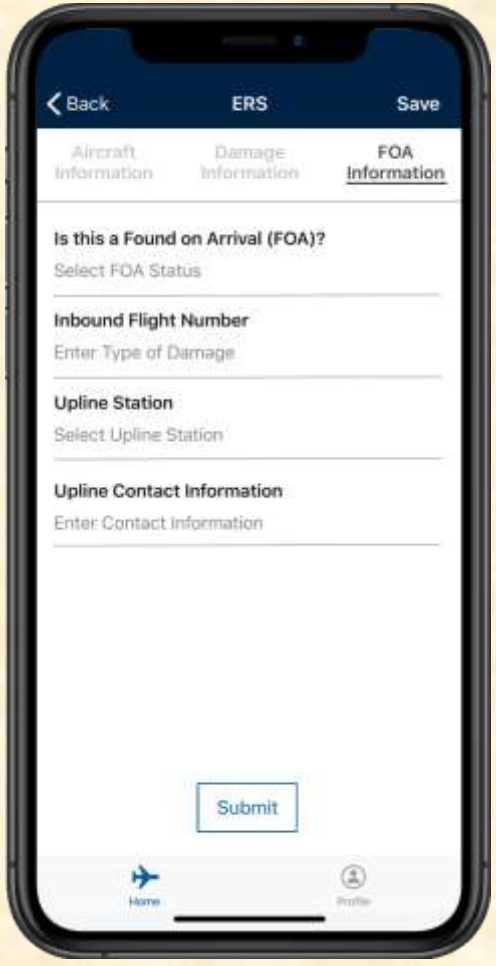


Screen Mockup: iOS MOCHA & LOSA Form



Screen Mockup: iOS ERS Form

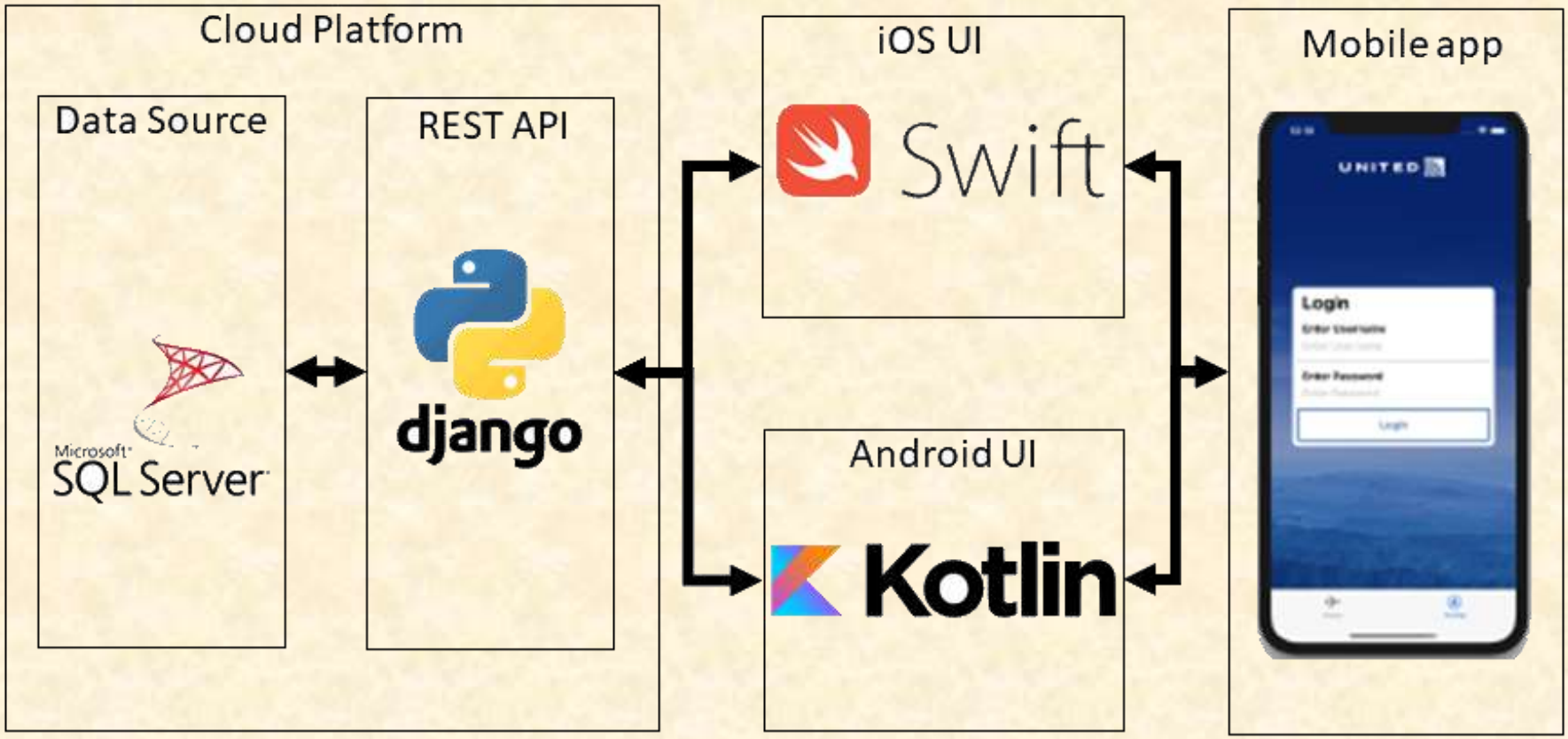




Technical Specifications

- This app will be designed to support both iOS and Android natively and follow interface design guidelines for both platforms.
  - Swift for iOS
  - Kotlin for Android.
- Based on their access levels, different employees will have access to different forms. The UI will automatically update at login to show only relevant audits and forms.
- The devices will communicate with our REST API endpoint build using Django.
  - The Django server will be responsible for querying and updating the MS SQL Server database provided by United Airlines.

System Architecture



System Components

- Hardware Platforms
  - Microsoft SQL Server
  - Web Service: Python and Django library to create REST API web service
- Software Platforms / Technologies
  - Xcode: Apple IDE developing software
    - Swift
  - Android Studio: Google IDE for creating Android applications
    - Kotlin

Risks

- **Risk#1: Connection to Backend Database/Sandbox-Integration**
  - Description: As access to database is still in progress, we are creating the front end side, however when we do get granted database access, connecting and integrating with frontend will be difficult, as the database they created for the mobile application is new and did not exist last semester. Also very large amounts of data that needs to be transferred.
  - Mitigation: Take an iterative approach, doing one feature at a time, and focus on how to integrate properly. By doing that, it allows room for trail and error.
- **Risk #2: Authentication of User and their access level**
  - Description: Based on the type of user that is accessing the mobile app, their access level and clearance will be different than other employees, and therefor only allow them to create certain reports or audits, not having access to all the types of reports.
  - Mitigation: Work with the IT team to understand the access levels and how they are determined. Also continue the building of last group's OAM (Oracle Access Manager) and make sure it stays secure and authenticates user when logging in to the mobile app.
- **Risk #3: IT and UX Support**
  - Description**:** With building and redesigning the mobile application, IT support from United is limited, due to COVID, and work schedules with vacation time, and Friday's off.
  - Mitigation: The best strategy to mitigate this problem is to ask a lot of questions when we do have meetings with them, and to repeatedly send them updates, document all work as clearly and detailed as possible, keep all work transparent, so when problems arise, it can be fixed or debugged more effectively.

Questions?