
C O
O P
UC COOPeration
Team COOPeration: Kartikey Pandey, Peter Kroeger, Domas Karvelis,
Ryan Kunkel, Nicholas McClorey

Introduction

What are we solving and how are we solving?

The Problem

- No official way to allow co-op students at the University of Cincinnati connect and share.
- Students are generally eager to learn from others' experiences, however, no easy or simple way to do so.
- Students are not aware of different company cultures due to lack of an established co-op community.

The Solution: UC COOPeration

- UC COOPeration is a social media platform to connect co-op students at the University of Cincinnati
- By having a space for students to connect about co-op, the overall co-op experience will be enhanced
- This application could help a student decide where they plan to do their co-op terms while at UC
- This will be accomplished by creating a backend server and database that supports the web and mobile application
- It will establish a co-op community at UC

Scope

This project will:

- Consist of:
  - A web app
  - An Android Application
  - A database
- Authenticate users
- Allow students in the co-op program to connect with other students
- Display information about students and alumni who have worked with various companies
- Allow users to create posts and share with connections

This project will not:

- Prevent non-UC students to register
- Encrypt messages between users
- Contact employers about job opportunities
- Gather current news
- Prevent users from creating multiple accounts with different email addresses

Assumptions & Constraints

- The only assumption made is the assumption that users will only utilize the app for its intended purpose
  - This allows the designers to not have an additional design requirement that was deemed unnecessary
- Three primary constraints
  - Development Time
    - The 11-week timeframe for the project requires the team to diligently produce the product
  - Remote Work
    - Lack of face-to-face interactions causes more difficulty communicating
  - User Willingness Needed for Success
    - For COOPeration to work, users must be willing to register and enter their information

Design Changes

Changes we can make to our product design

What worked?

- Using Firebase for the backend, helped us save a lot of time in implementation
- The Model-View-Controller architecture allowed both the apps to be in sync
- Designing the UI in XD at the start helped us visualize the app better for implementation

What did not work?

- Functionality of user posts, which would store either text or images as well as comments and likes, was not achievable
- The idea of user notifications was removed in favor of simply showing new messages and connections at the top of their respective page
- Originally, for testing we planned on having real users create accounts, but real user testing was never accomplished

Test Cases

Test Cases and their execution

Test Cases

- T001 – Feed Posts

Date	Tester	Pass/Fail	If fail, issue	Comments
08/07/20	Kartikey	Fail	Implementation	Posting feed couldn't be implemented because of the time constraint

- T002 – Account Creation

Date	Tester	Pass/Fail	If fail, issue	Comments
08/07/20	Kartikey	Pass		Works perfectly, no bugs

Test Cases

- T003 – User Connection

Date	Tester	Pass/Fail	If fail, issue	Comments
08/07/20	Ryan	Pass		

- T004 – Sending a Chat message

Date	Tester	Pass/Fail	If fail, issue	Comments
08/07/20	Kartikey	Pass		Works perfectly, no bugs

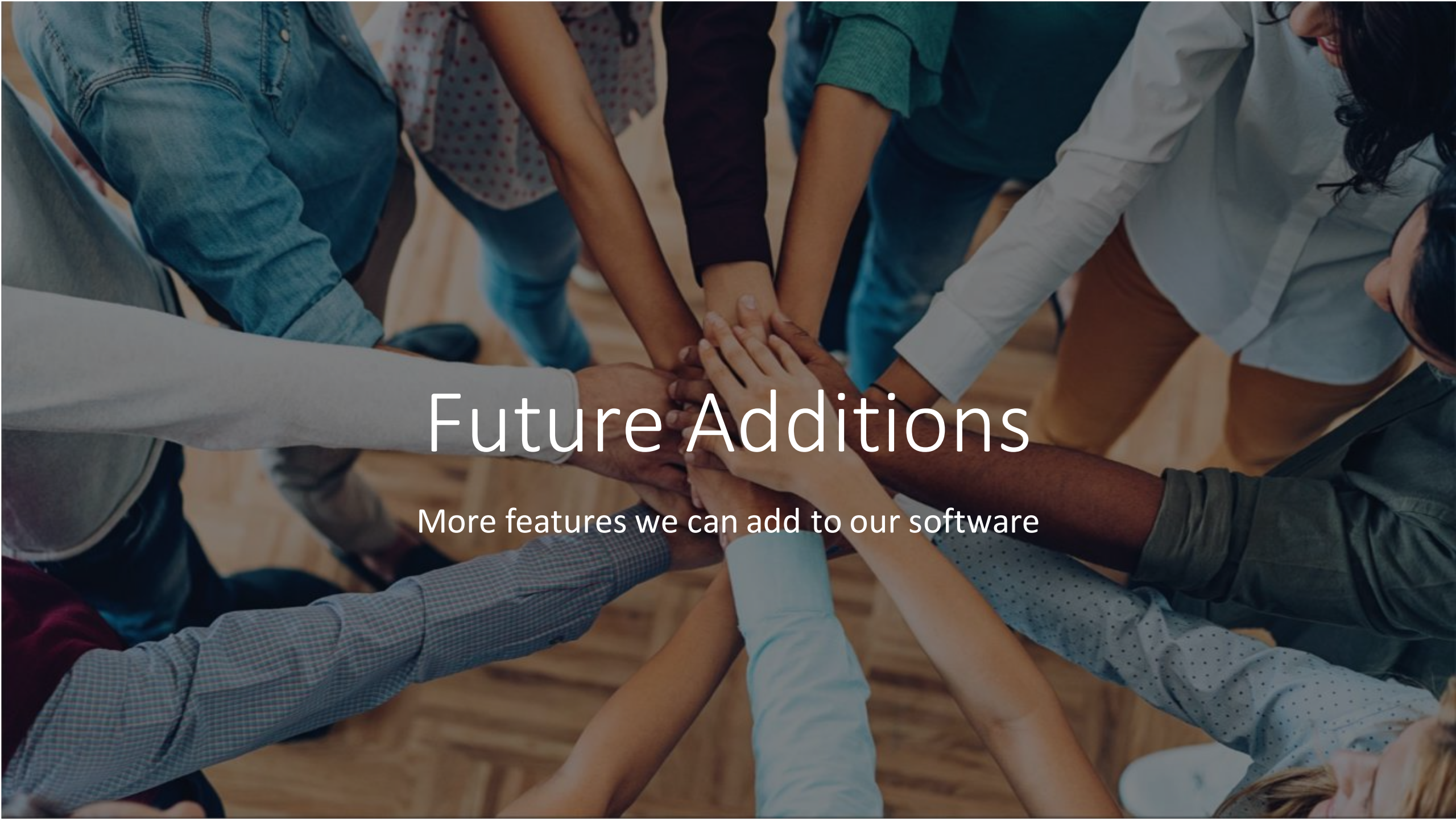
Test Cases

- T005 – Company History

Date	Tester	Pass/Fail	If fail, issue	Comments
08/07/20	Ryan	Pass		Works perfectly, no bugs

- T006 – User Search

Date	Tester	Pass/Fail	If fail, issue	Comments
08/07/20	Nick	Pass		Works perfectly, no bugs



Future Additions

More features we can add to our software

Potential Features

- Implement a "Suggested Connections"
- Multiple pictures in a single post
- Posts with video
- Privacy Settings
- Sign-up with LinkedIn

Final Conclusions

Technical

- Working with Firebase was easy because of the provided documentation and examples
- Using Visual Studio Code for the web app proved to be useful as the web app team used it in the past for development
- Developing the Android App using Kotlin was new for the app developer, it had its own learning curve

Process

- Azure DevOps worked well to organize COOPeration
- Break up the sections of the application to focus on weekly
- Weekly reviews of code changes from the team
- Concise weekly goals would help to stay on track

Group Dynamic

- Keeping track of team members' progress
- Holding more meetings beyond documentation meetings
- Team members completing work by the time stated
- People reaching out for help when needed


Thank You