CPSC 411

GUESTBOOK

EMILY CROWL

Section 1: Functionalities

Required Must-have Stories:

- ➔ Users can create an event
  - ◆ Users can create an event in the application using their email and a password of their choice
- ➔ Users can join an event
  - ◆ Users can join an event in the application using their email and the event password
- ➔ Users can edit event details
  - ◆ Currently only displays form
- ➔ Users can post a picture/video/text message
  - ◆ Currently nonfunctional
- ➔ Users can view a feed of media
  - ◆ Currently hardcoded data
- ➔ Users can view information about the app and get help
  - ◆ Terms and Privacy Policy, currently links to Lorem Ipsum, as that information has not yet been generated for this application

Section 2: Architecture & Design

Database Schema

Property	Type	Description
objectId	String	Unique id for the user post
author	String Pointer to User	author's username
eventId	String	Unique id for the event
image	File	image that user posts
caption	String	image/video caption by author
video	File	video that user posts
createdAt	DateTime	date when post is created
updatedAt	DateTime	date when post is last updated

Third-Party Libraries

- ➔ CocoaPods
- ➔ Appirater
- ➔ SDWebImage
- ➔ Crashlytics

Section 4: GitHub Repository

https://github.com/fooledyouonce/GuestBook_iOS

Section 5: Test Results

- ➔ Scenario 1: Unregistered User
  - ◆ *Data:* email = test@gmail.com, password = password
  - ◆ *Successfully completed?:* Yes, user was not able to log in
- ➔ Scenario 2: Registered User, valid data
  - ◆ *Data:* email = test@gmail.com, password = password
  - ◆ *Successfully completed?:* Yes, user was able to log in
- ➔ Scenario 3: Registered User, invalid password
  - ◆ *Data:* email = test@gmail.com, password = passworf
  - ◆ *Successfully completed?:* Yes, user was not able to log in
- ➔ Scenario 3: Registered User, invalid email
  - ◆ *Data:* email = tesy@gmail.com, password = password
  - ◆ *Successfully completed?:* Yes, user was not able to log in
- ➔ Scenario 4: No data entered
  - ◆ *Data:* email = nil, password = nil
  - ◆ *Successfully completed?:* Yes, user was not able to log in
- ➔ Scenario 5: Blank field, email
  - ◆ *Data:* email = nil, password = password
  - ◆ *Successfully completed?:* Yes, user was not able to log in
- ➔ Scenario 6: Blank field, password
  - ◆ *Data:* email = test@gmail.com, password = nil
  - ◆ *Successfully completed?:* Yes, user was not able to log in
- ➔ Scenario 7: Registered User, not a valid email (no @)
  - ◆ *Data:* email = testgmail.com, password = password
  - ◆ *Successfully completed?:* Yes, user was not able to log in
- ➔ Scenario 8: Registered User, not a valid email (no .)
  - ◆ *Data:* email = test@gmailcom, password = password
  - ◆ *Successfully completed?:* Yes, user was not able to log in
- ➔ Scenario 9: Registered User, not a valid email (no @ and .)
  - ◆ *Data:* email = testgmailcom, password = password
  - ◆ *Successfully completed?:* Yes, user was not able to log in

Section 6: Future Work

Functionalities & Features:

- ➔ User login screen
  - ◆ Users can log into the application with a username/email and password
- ➔ User registration screen
  - ◆ Users can create an account with their email, username, and password
- ➔ Navigation to join event/create event screen from login screen
  - ◆ From the login screen, the application will navigate to a screen with join/create event options. They will also have access to their profile.
- ➔ Post creation
  - ◆ Users can create a post to post onto the event timeline. Posts can be photos, videos, text, and/or drawings.
- ➔ Post feed
  - ◆ Users can view a feed of posts by different users for a given event
- ➔ View past events
  - ◆ Users can view the past events that they have joined/created
- ➔ Comment on posts
  - ◆ Users can comment on posts and comments are displayed for all users within a given event
- ➔ Like posts
  - ◆ Users can like posts and likes are displayed for all users within a given event
- ➔ View live notifications
  - ◆ Users get notified when someone interacts with their post (likes, comments)
- ➔ View a live gallery of event images
  - ◆ Users can view a gallery of all posts for a given event
- ➔ Edit profile
  - ◆ Users can edit and update their profile

Enhancement of Architecture & Design:

- ➔ UX
  - ◆ Improve for a more polished user experience
- ➔ Interconnections (API)
  - ◆ Improve security and data management
- ➔ Security
  - ◆ Implement a more secure system for storing users' personal information
  - ◆ Generate random, unique passcodes for events
- ➔ Network (bandwidth, latency)
  - ◆ Ensure application establishes an internet connection and takes up minimal bandwidth
- ➔ Data types (entities) and relationships
  - ◆ User → ensure unique username and email
  - ◆ Event → ensure unique passcode
  - ◆ Log in, join event, view posts → fetch data from database
  - ◆ Sign up, create event, upload posts → add data to database
- ➔ Storage (latency, capacity, permanence)
  - ◆ Database is able to handle large amounts of data
  - ◆ Fetch time is not slow
- ➔ Performance
  - ◆ Can handle large amounts of simultaneous users
- ➔ Component/Layers
  - ◆ Database → fully develop
  - ◆ Front end → improve UI/UX
  - ◆ Backend → ensure data retrieval and persistence works
- ➔ Accessibility
  - ◆ High contrast mode
  - ◆ Looks polished in both light and dark mode
  - ◆ Partial blindness accessibility (read aloud features, image descriptions)
  - ◆ Easy-to-use interface
  - ◆ Generate and toggle captions on videos