admon: task alert system

Class: University of Colorado - Boulder: CS 3308

Professor: David Knox

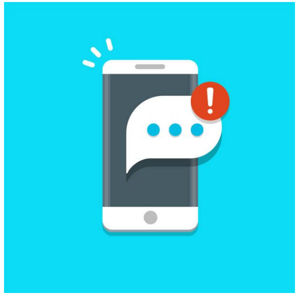
Team 1: Alex Sueppel, Cole Gaito, Jake Sandelin, Kyle Tomlinson

Purpose of admon

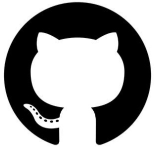
The purpose of admon is to allow for easy notification systems to a users phone via text message. Alarms and notifications can be customized to fit user preferences of **daily, weekly, or one time notifications.**

Functionality:

- Email notification would be sent to the specific referenced number.
- Notification system would check every 5 minutes to see if additional alarms needed to be sent out.



Tools Used



Tools Used: Project Management

Trello:

Pros:

- Easy to add notes and comments
- Could duplicate cards to allow for assignment and categorization

Drawbacks:

- Difficult to engage other uses if the individuals did not access the board
- Did not seem to have external notification, such a email

Github:

Pros:

- Manages merging and file retention
- Easy to pull data when you were going to work on the project.

Drawbacks:

- Occasionally disjoining
- Failure to set-up project correctly can result in merging errors and redundant work
- When pulling the information did not necessary see other's comments

Tools Used: Development

HTML/CSS/JavaScript:

Pros:

- Web standard
- Extensive documentation
- Multiple avenues for problem solving

Drawbacks:

- Unpredictable across browsers
- Limited experience within group, frontend developers learned as they worked.
- Choice paralysis: ability to create the same result across three languages complicated decision making

Python:

Pros:

- Easy to learn
- Extensive documentation
- Large community around problem solving
- One of the better languages for data management

Drawbacks:

- Many different versions that highly affect outcome of code
- Limited to backend and database management. Not a tool recommended for full-stack development by itself compared to JavaScript.

Tools Used: Development

VS Code:

Pros:

- Familiarity with editor
- Easily be able to transition between Terminal and files for testing and development
- Integration with GIT if you use add-ons

Cons:

- Many tools exist in system and can be overwhelming
- A bit of a walled garden, even though it is an open sourced project.

SQLITE3:

Pros:

- Very easy to develop in for simple projects
- Lightweight and somewhat self contained
- Native support included in Python3

Cons:

- Scalability issues

Tools Used: Development

Flask:

Pros

- Widespread prevalence
- Relatively simple to use
- Highly modular

Cons

- Difficult to run in a remote environment
- Modularity only advantageous if developers are aware of existing modules

Heroku:

Pros

- Lightweight system
- Lots of support online
- Free system that has a decent amount of scalability before needing to pay

Cons

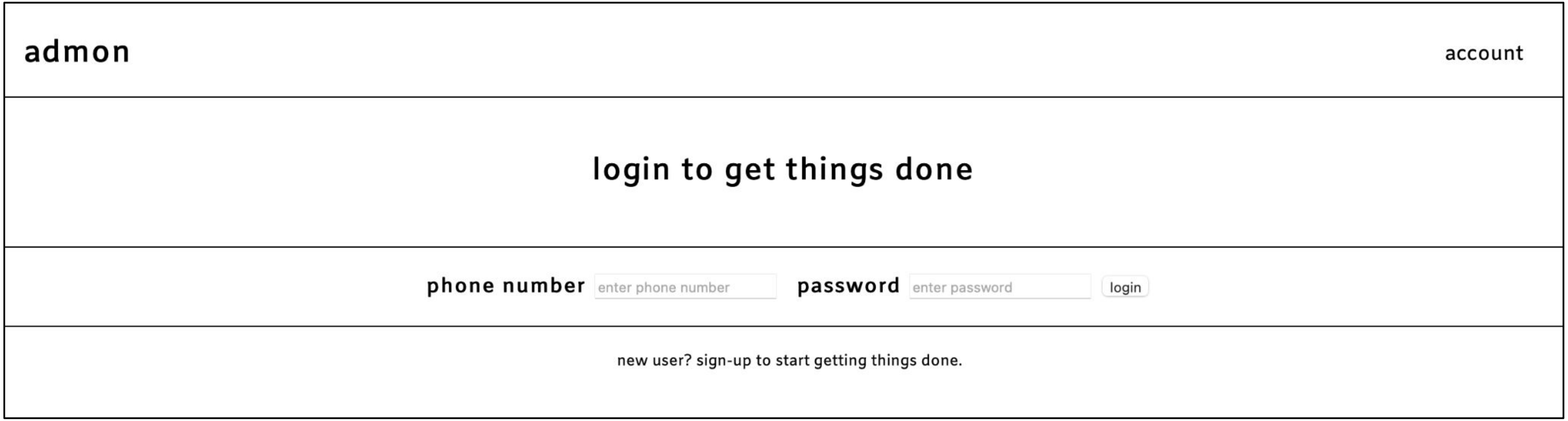
- Confusing at times and doesn’t play nicely with many versions of other tools
- Requires very specific python and git versions
- Cannot be implemented effectively half way into project development.

Testing

Database/Backend:

- Provided individual and customized testing for system using VSCode to monitor database on insertion and modification of code via terminal prompts and a main function.
- Individual manual testing was set up, as opposed to a common testing framework due to limited resources and allocation of time. The developer was the main tester of their own code, and collaboration provided correction of missing pieces.

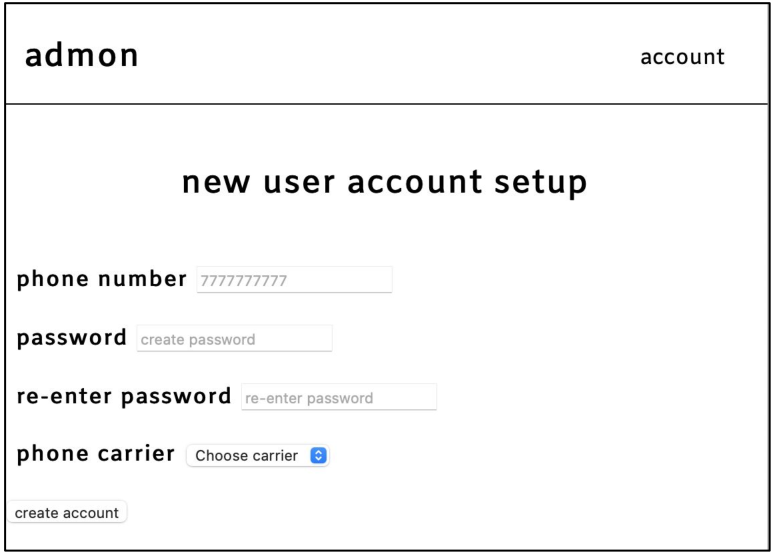
User Interface: Login Page



login to get things done

new user? sign-up to start getting things done.

User Interface: New User & Task Creation



admon

account

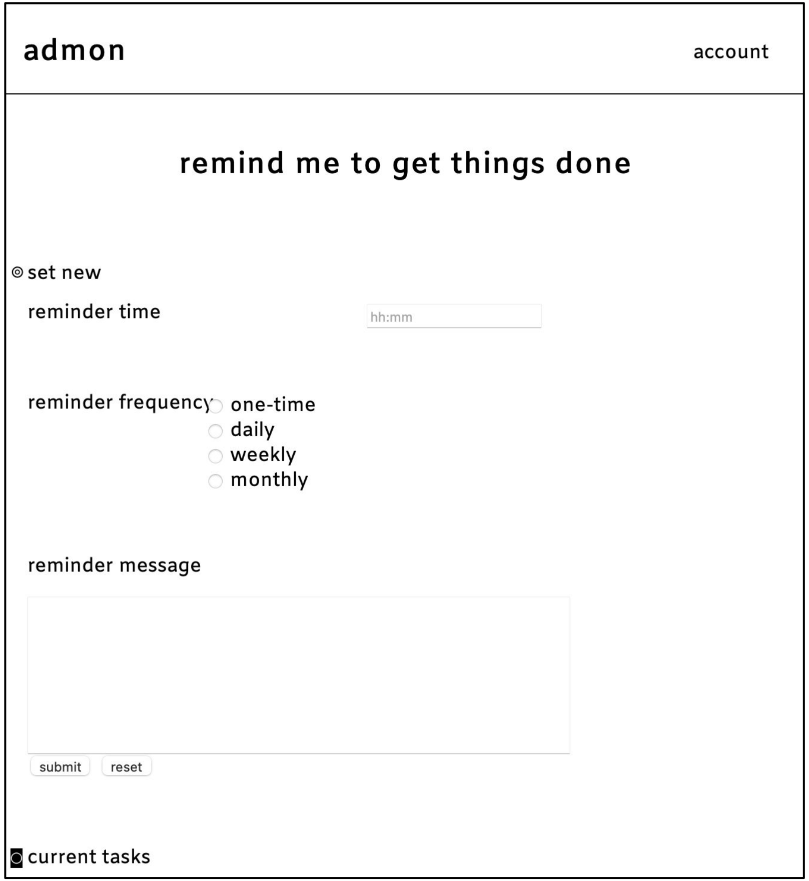
new user account setup

phone number 7777777777

password create password

re-enter password re-enter password

phone carrier Choose carrier



admon

account

remind me to get things done

◉ set new

reminder time hh:mm

reminder frequency
○ one-time
○ daily
○ weekly
○ monthly

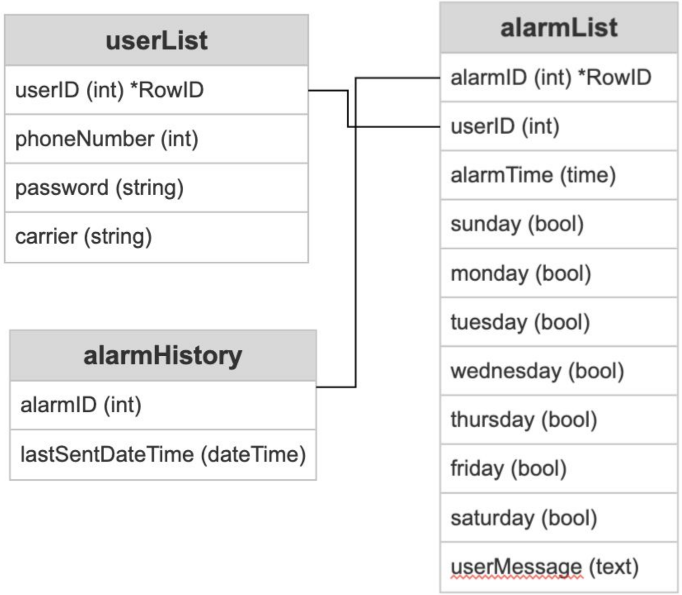
reminder message

● current tasks

Database:

Design

- Simple relational database design
- Utilizing unique rowID's automatically generated in SQLite3
- Easy scalability potential and growth as stretch goals were achieved for product
- Established to work on a weekly message system with clear path to grow into:
    - One time messages
    - Monthly Messages
    - Other Unique Requirements



userList
userID (int) *RowID
phoneNumber (int)
password (string)
carrier (string)

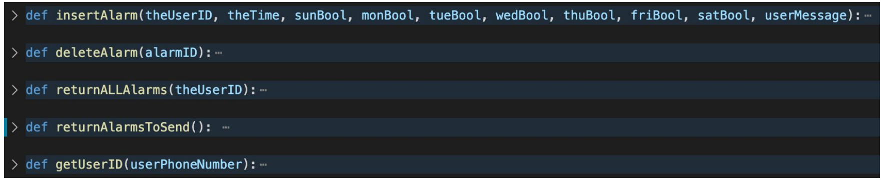
alarmHistory
alarmID (int)
lastSentDateTime (dateTime)

alarmList
alarmID (int) *RowID
userID (int)
alarmTime (time)
sunday (bool)
monday (bool)
tuesday (bool)
wednesday (bool)
thursday (bool)
friday (bool)
saturday (bool)
userMessage (text)

Back-End: Black Box Environment

Concept:

- Functions that work with the database were designed to be easy to use without the need of other developers to need to understand every piece of the data environment.
- Allowed for quick customization and implementation of systems in the project to interact with the database via parameter arguments.
- Example:



```
> def insertAlarm(theUserID, theTime, sunBool, monBool, tueBool, wedBool, thuBool, friBool, satBool, userMessage):⋯

> def deleteAlarm(alarmID):⋯

> def returnALLAlarms(theUserID):⋯

> def returnAlarmsToSend(): ⋯

> def getUserID(userPhoneNumber):⋯
```

Challenges

- Producing individual deliverables on time
- Integrating different project aspects (frontend, database, etc.) near end of development
- Scheduling meetings with all members (consistent attendance)
- Keeping meetings on time and concise

Solutions Going Forward

- Prioritize the reduction of larger goals into smaller, deliverable tasks
  - This is vital to the success of agile processes and helps ensure that developers can meet goals on a weekly basis
- Integrate early and often
  - Reduces likelihood of major integration issues at end of development
- Dedicate more attention to project management system
  - Ensures developers maintain an understanding of where they are in the project timeline
- Commit to fixed sprint periods
  - Promotes developer accountability and encourages better time management

Thank you!

From the admon team