Plantam

Group Project Presented by
GROUP 4

Brian Cieplicki

Isabella Dahilig

Bemnet Demere

Anh Nguyen

Binh Vo

Be a bantam. Plan it.

Plantam

Problem Statement

- Decentralized communication
- Current platforms for communication:
  - TrinityToday for general events
  - Facebook for cultural and social events
  - Emails for academic events

Problem Statement

- Plantam is designed for an individual of Trinity college to easily access public campus events and plan personal events along with it

BackLog Features

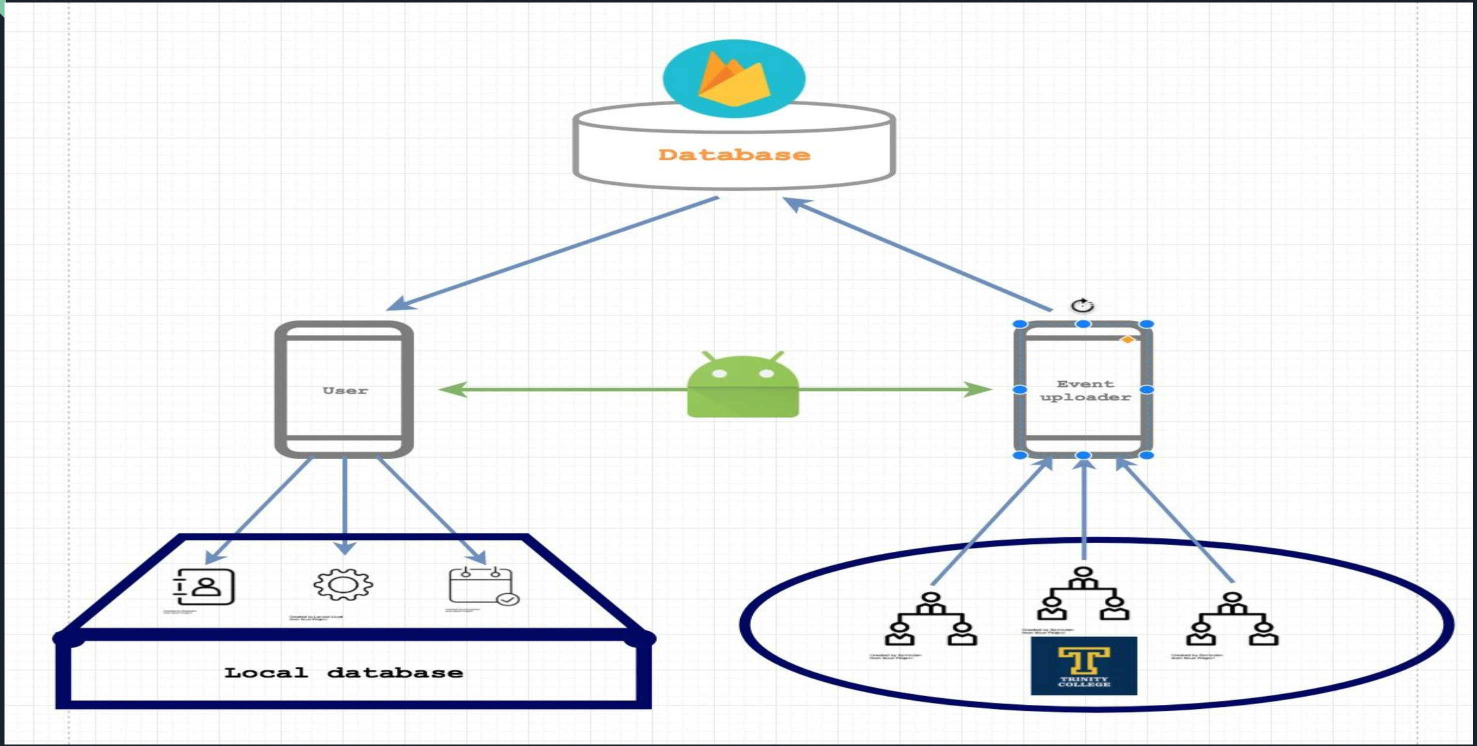
- High-Priorities:
  - Each user can have a private account.
  - View public events
  - Users can add, modify, and delete personal events
  - Add public events to personal calendar
- Low-Priorities:
  - Create an account for new user
  - Display location of event
  - Remind user of an upcoming event

Demo

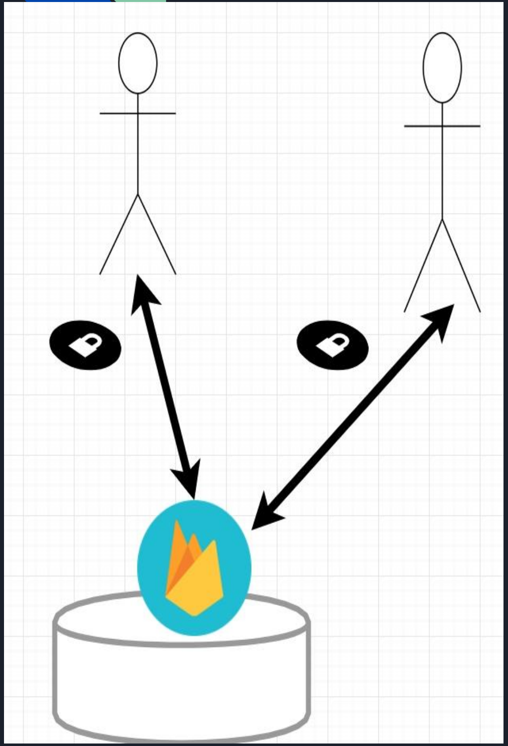
Technologies

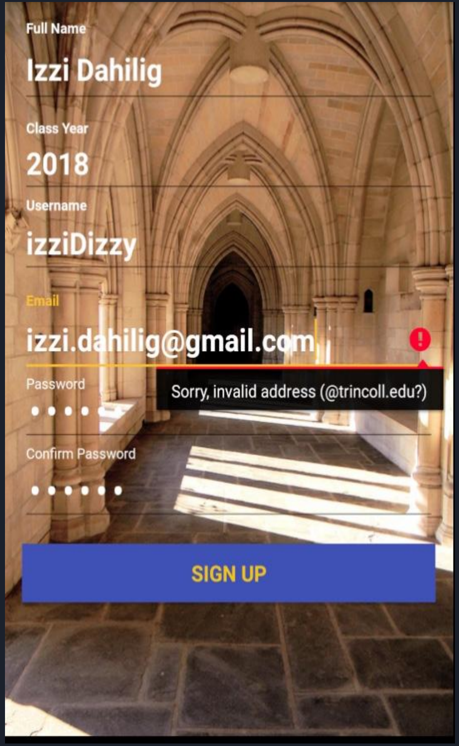
- Android Studio
- SQLite Database
- FireBase
  - Authentication
  - Storage (Events and User Info)
- Google Mobile Services
- Jsoup

Approaching our Designs....



Sign Up and Authentication





Full Name

Izzi Dahilig

Class Year

2018

Username

izziDizzy

Email

izzi.dahilig@gmail.com

Sorry, invalid address (@trincoll.edu?)

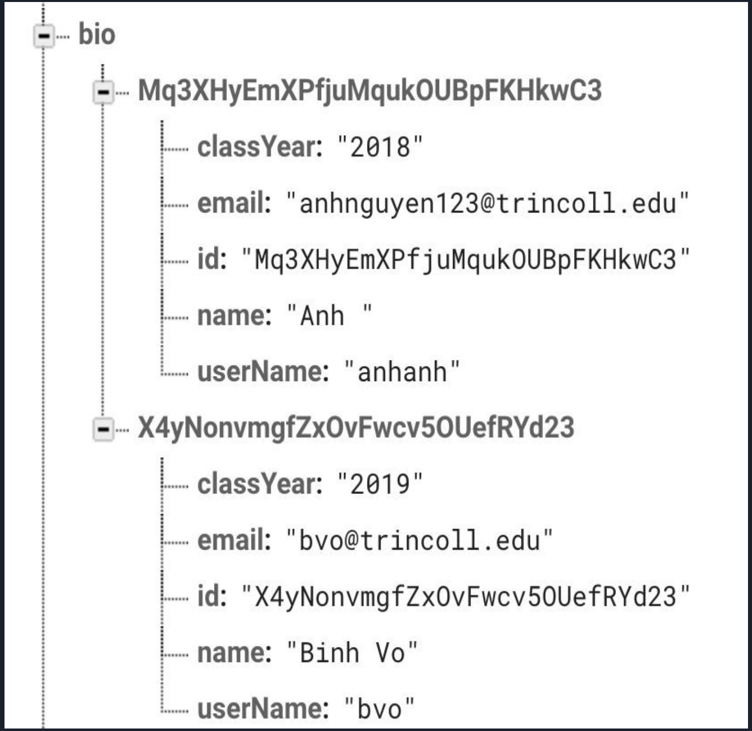
Password

•••••

Confirm Password

••••••

SIGN UP



```
bio
  Mq3XHyEmXPfjuMqukOUBpFKHkwC3
    classYear: "2018"
    email: "anhnguyen123@trincoll.edu"
    id: "Mq3XHyEmXPfjuMqukOUBpFKHkwC3"
    name: "Anh  "
    userName: "anhanh"
  X4yNonvmgfZxOvFwcv5OUefRYd23
    classYear: "2019"
    email: "bvo@trincoll.edu"
    id: "X4yNonvmgfZxOvFwcv5OUefRYd23"
    name: "Binh Vo"
    userName: "bvo"
```

Encapsulate what varies

- Encapsulate event objects for
    - Firebase and SQLite
    - CustomDate, CustomTime

Principle of Least Knowledge

- Our classes do not know any unnecessary information about other classes.
    - DatabaseHelper ->SQLiteDatabse

Observer Pattern

- Firebase Database
- Hollywood Principle
- Allows classes dependent on Firebase Database to stay updated on the information in the Database without repeatedly asking for information
- Makes the application more efficient on user side
- Creates ‘subscriber’ relationship between user and Database

Template Method Pattern

- 2 classes two consider
- Both have similar types of roles
- One is to add an event
- And the other is to retrieve and edit event (thus adds the event)

Facade Pattern

- The FBDatabase class acts as a facade for the app
- It provides an encapsulation so that the application that uses Firebase's routines need not know about the technicality of the platform

Questions?

Thank you!