

A black square icon with the text "Space: 9+" in white, bold, sans-serif font.

Space: 9+

CLASS SPACE MONITOR APP

An iOS app for Northeastern student
to monitor class remaining space

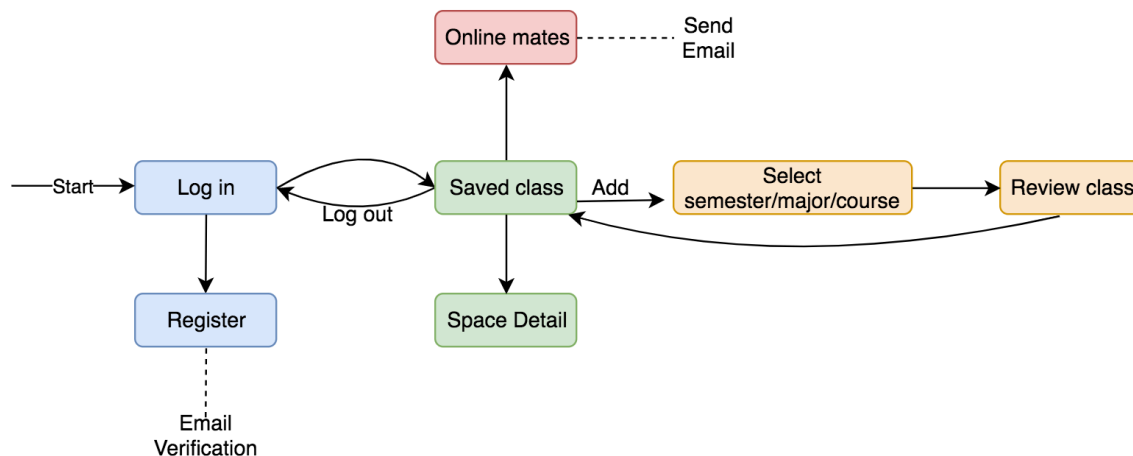
Zhiyi
Wang

Introduction

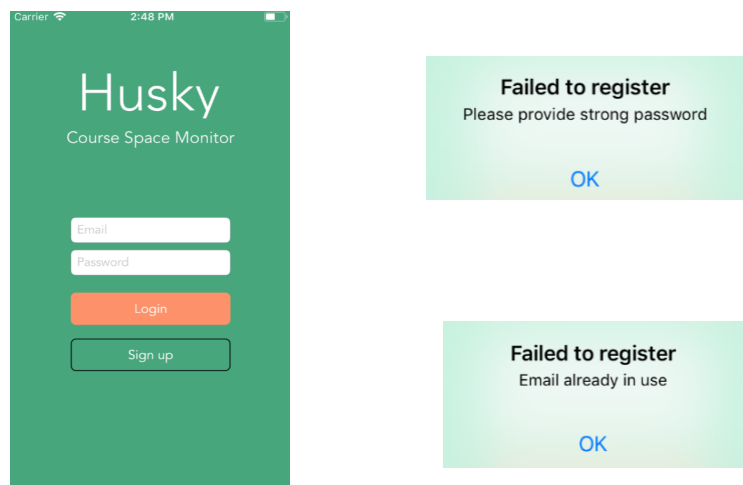
My final project is an iOS application specifies for the Northeastern University students to monitor their class registration available space. The application allows current Northeastern student, registration by email, login once, then conveniently monitor the class they desire to register, in terms of seats opening and waiting list status. The project's client side is implemented by Swift, with functions of account register, login, checking online school mates. Once student logged in with his or her account, the student can add/delete up to 10 Northeastern courses for monitoring purpose. The server side is implemented by Firebase and a web crawler.

Application Workflow

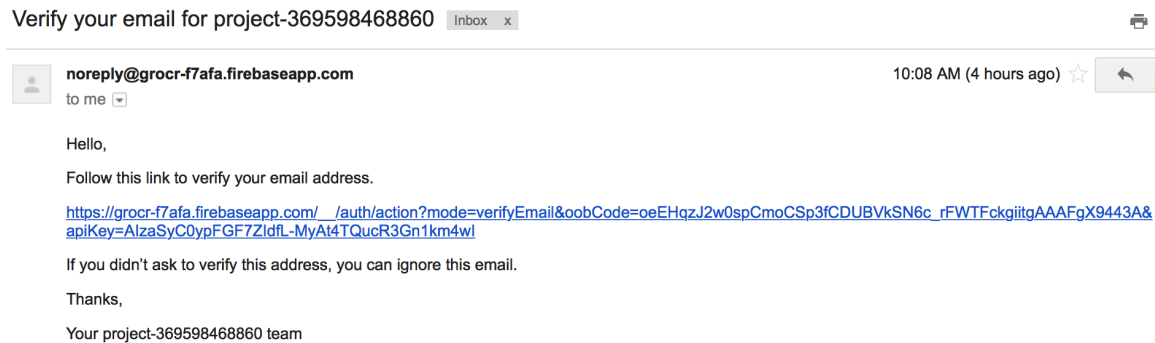
The general flow for app is that, user can log in, then check saved class space to monitor on home page, with functionalities of adding more class to monitor as well as check who are also online using this app for communication purpose.



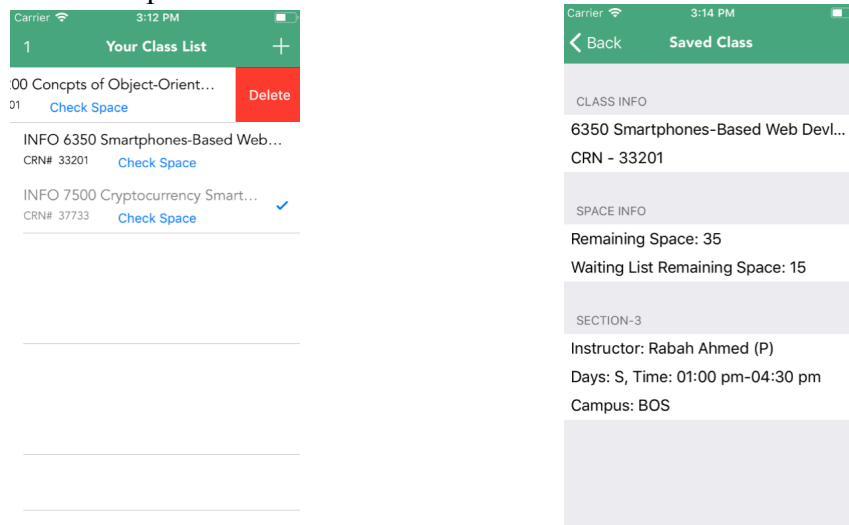
The application starts from the log-in page, so each user can register their account by an email address, with a valid password. The invalid registration, including weak password or existed email account, will be alerted to user and the registration won't be successful.



After successful registration, an account verification email is sending to the user's email address at the same time. User will need to verify her or his account through clicking the link in email.



With successful log-in, user can see all his/her saved class to monitor. A checkbox has been provided at the right side of each cell for user's ease. User can click "Check Space" for space information and other details. To delete your saved class, just drag the cell to the left, and an 'delete' button will show up.



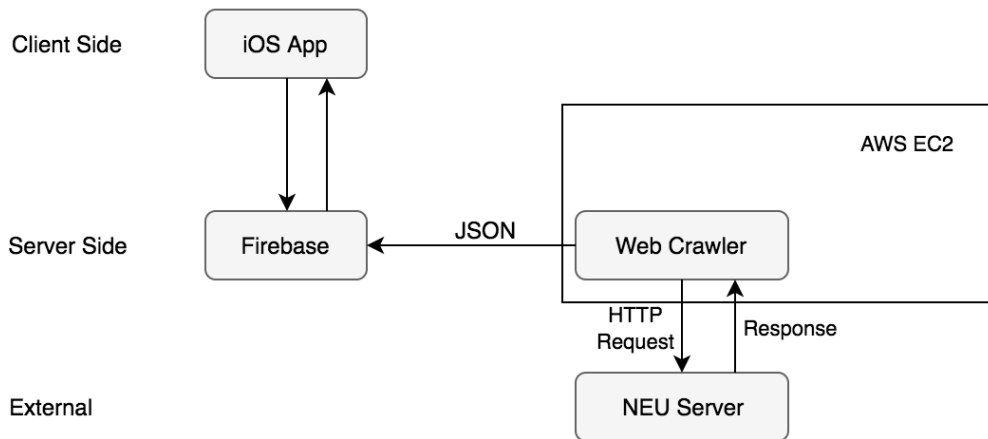
The left top corner button is a number showing how many people are online using this application, and user can click this number and send email corresponding to a certain online account email address. The right top corner plus button is for adding new classes to monitor, by selecting semester, major, course, and class section.



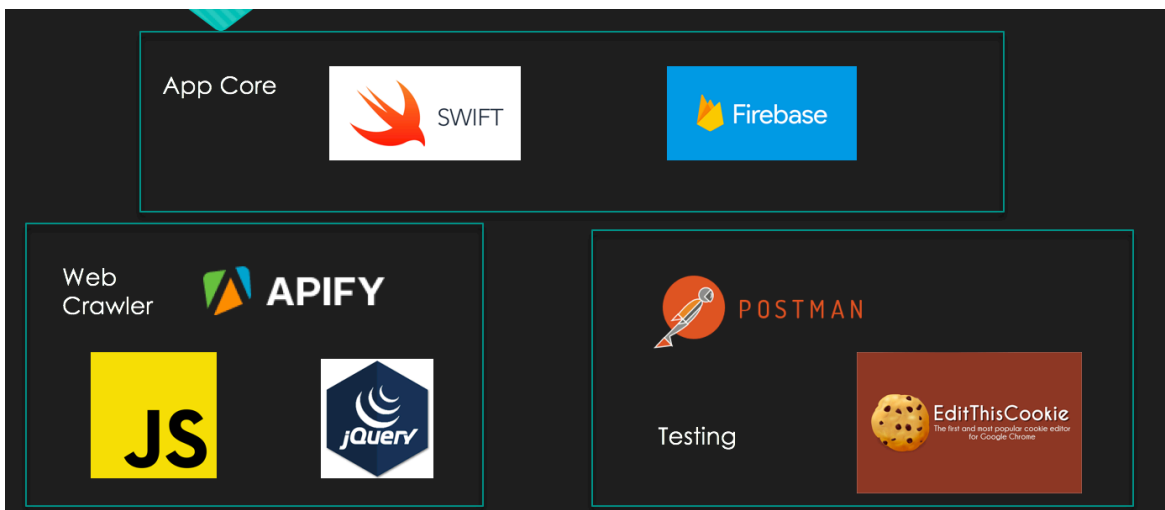
Right now, the project only sampled 'Spring 2018' semester for CSYE or INFO major classes for testing purpose by web crawler, full class list will be sampled later for full version application.

Software Structure

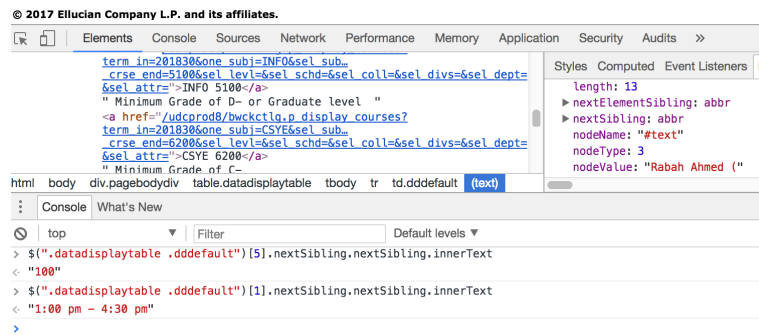
The software structure consists of front-end by iOS app in Swift, back-end by Firebase database and a web crawler to update our data from Northeastern server constantly. The web crawler sends HTTP requests with either GET/POST method to Northeastern University server with certain Semester and CRN values (as body for POST method, and in URL link when as GET method), then it catches web page content by jQuery selector and capsulate it into JSON format data, which the JSON data will be exported from my web crawler and import into Firebase. Then, based on Firebase database, iOS app can fetch all the course and class space info and details.



Many technologies involved for making this software come true. Not only Swift 4 with Xcode and Firebase database implemented for data adding/deleting/updating/searching, but also much amount of work was put into Web Crawler building and testing.



After many days research and self-learning, the APIFY, a web scraping and automation platform, was chosen to achieve the web crawler function to extract contents from the loaded page. Firstly, by listening Northeastern Dynamic Schedule, the useful HTTP requests list were collected. Secondly, by carefully checking HTML elements, I used jQuery selector to detect useful information and formatted it into JSON data by JavaScript functions. Lastly, import the JSON data into Firebase.



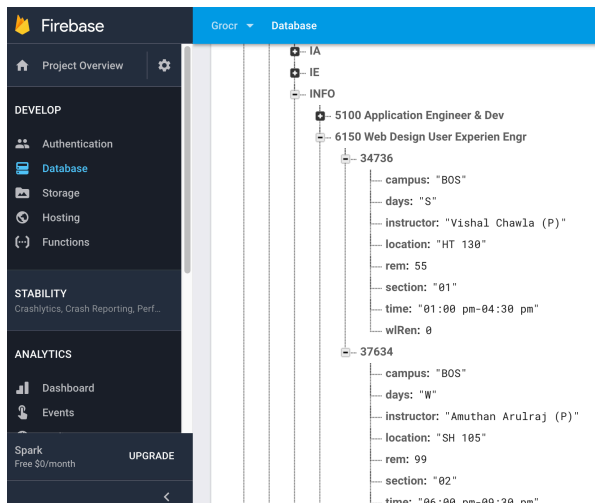
Elements inspect in Chrome

```
function pageFunction(context) {
    // called on every page the crawler visits, use it to extract data from it
    var $ = context.jQuery;
    var result = {
        "Course": $("".ddtitle").text().trim(),
        "Associated Term": $("".dddefault.fieldLabelText")[0].nextSibling.data.trim(),
        "Levels": $("".dddefault.fieldLabelText")[1].nextSibling.data.trim(),
        "Attributes": $("".dddefault.fieldLabelText")[2].nextSibling.data.trim(),
        "Instructors": $("".dddefault.fieldLabelText")[3].nextSibling.data.trim(),
        "Schedule Type": $("".dddefault.fieldLabelText")[4].nextSibling.data.trim(),
        "Time": $("".datadisplaytable.dddefault")[1].nextSibling.nextSibling.innerText,
        "Days": $("".datadisplaytable.dddefault")[2].nextSibling.nextSibling.innerText,
        "Where": $("".datadisplaytable.dddefault")[3].nextSibling.nextSibling.innerText,
        "Date Range": $("".datadisplaytable.dddefault")[4].nextSibling.nextSibling.innerText,
        "Capacity": $("".datadisplaytable.dddefault")[5].nextSibling.nextSibling.innerText,
        "Actual": $("".datadisplaytable.dddefault")[6].nextSibling.nextSibling.innerText,
        "WL Act": $("".datadisplaytable.dddefault")[7].nextSibling.nextSibling.innerText,
        "Seats": $("".datadisplaytable.dddefault")[8].nextSibling.nextSibling.innerText,
        "Room Size": $("".datadisplaytable.dddefault")[9].nextSibling.nextSibling.innerText
    };
    return result;
}
```

Web Crawler selector function by jQuery/JS

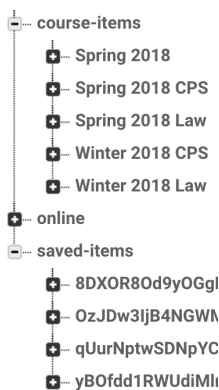
```
[{"Course": "Smartphones-Based Web Development - 33201 - INFO 6350 - 01- (Boston College)", "Associated Term": "Spring 2018 Semester", "Levels": "Graduate", "Attributes": "GSEN Information Systems", "Instructors": "Rabah Ahmed (", "Schedule Type": "Lecture", "Time": "1:00 pm - 4:30 pm", "Days": "S", "Where": "Shillman Hall 305", "Date Range": "Jan 08, 2018 - Apr 28, 2018", "Capacity": "100", "Actual": "65", "WL Act": "0", "Seats": "35", "Room Size": "114", "url": "https://wl11gp.neu.edu/udcprod8/NEUCLSS.p_class_search?sel_day=dummy"}]
```

JSON formatting and export by web crawler



Firebase final data

For Firebase, it provides the software with Authentication and Database functions. The authentication allows our user to Register/Log-in our app, and the database allows data persistence of class space info, saved class info, and online user info.

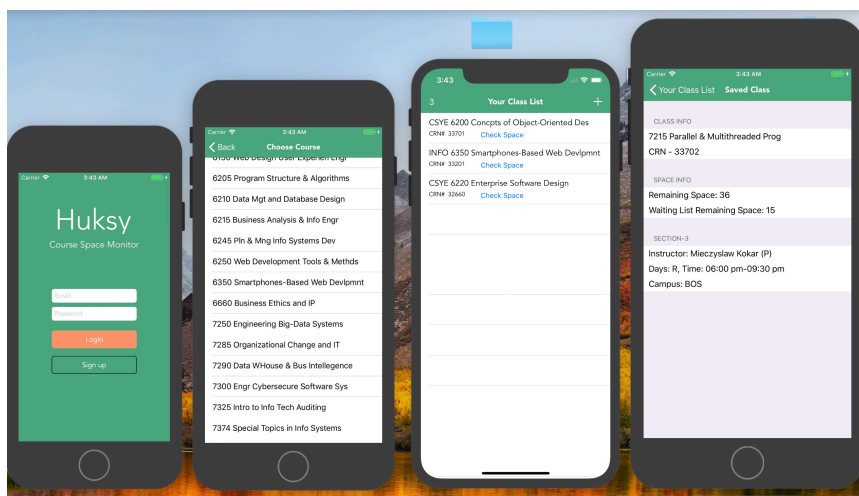


Database structure

Identifier	Providers	Created	Signed In	User UID ↑
wangzhiyi05@gmail.com	📧	Dec 1...	Dec 1...	8DXOR8Od9yOGgl
test0001@gmail.com	📧	Dec 1...	Dec 1...	OzJDw3ljB4NGWM
test0002@gmail.com	📧	Dec 1...	Dec 1...	qUurNptwSDNpYC
jakewangstudio@gmail.com	📧	Dec 1...	Dec 1...	yBOfd1RWUdiMli

User authentication

Thanks to the auto-constraints in swift, the responsiveness never so good for all different screen resolutions and devices. All different iOS devices and screen sizes are fully tested and simulated.



Future Plan

The current application, is very much DEMO version, and long-term development with final release on Apple Store are planned for the month after the Christmas. At the beginning, it will work on letting users authenticate with Firebase using their Facebook/Google/other social accounts by integrating external SDKs or APIs Login into the app. Secondly, I will develop full jQuery and JavaScript function for full class list data. After that, online mates instant chat and push notification for new space of classes will be implemented for better user experience. With these basic features, I will be confident to launch and release my app onto Apple Store as Version 0.5. Moreover, a big progress will be made, which is to automate Web Crawler on AWS as EC2 instance and be running hourly, to update Firebase data and ensure the space info is timely. At the end, final trim will be done, and advertising at Northeastern campus will be promoted (Posters/ Facebook page/ Twitter and potential school clubs).

Time line	Action
○ Day 1 – 3	Facebook, Google, GitHub Login
○ Day 4 – 7	Web Crawler for all classes
○ Day 8 – 12	Online Mates Instant Chat + Push Notification
○ Day 13 – 15	App Store Release - Version 0.5
○ Day 16 – 20	Crawler Automation as RESTful API
○ Day 21 – 25	Deploy automated Crawler API to AWS
○ Day 26 - 29	App Final Trim & Better UI – Version 1.0

Thank you very much; I truly hope you will enjoy my app and I am excited to launch it on Apple Store next spring.