MMBS Cycle 1 Presentation

By Austin Mongold, Austin Newkirk, Chase Dumbacher and Sarah Pham

Cycle Intent

Our team's intent for this spike is to:

- Get the app running smoother than it was
- Verify previously implemented functions
- Create a PDF builder and automated emailer
- Make the app more intuitive
- Create basic website framework
- Implement firebase login for authentication
- Show firebase data in appropriate locations
- Create calendar display

Future Work

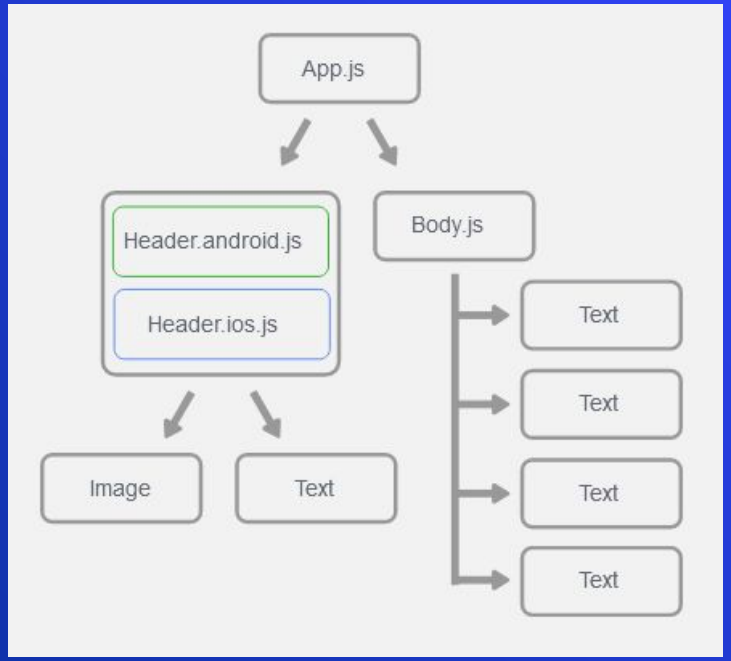
- Full implementation of emailer on mobile app
- Verify and improve PDF builder
- Fix rest of errors left by previous group
- Implement calendar with database information on website
- Improve data readability on website
- Add the ability to add, edit, and delete data from firebase database in consistent manner to mobile app
- Implement functional pdf builder and emailer similar to mobile app

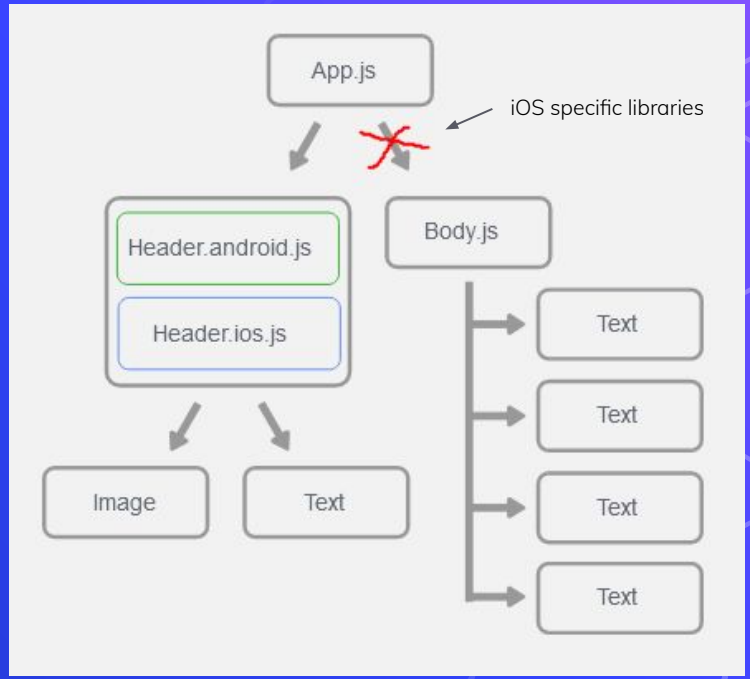
Management Plan

Cycle 1	Task owner	Status	Task Timeline	Priority	Dependent on	Trello Card
End-End Testing on general iOS app functionality		Stuck	Feb 5 - Mar 1	Medium	-	
Fix page navigations		Done	Feb 5 - 12	Medium	-	
Fix wrong homepage cache issues		Done	Feb 5 - 12	Medium	-	
Test event document generator		Done	Feb 3 - 12	Low	-	
Test event documents emailer		Stuck	Mar 2 - 8	Medium	Fix emailer auth issues	
Fix emailer auth issues		At risk	Feb 15 - Mar 1	High	-	
Fix iOS app dependency issues		Done	Feb 5 - Mar 1	Medium	-	
Create webpage frame layout		Done	Feb 5 - 19	Medium	-	
Setup firebase auth login		Done	Feb 5 - 12	High	-	
Adjust design per sponsor feedback		Done	Feb 5 - Mar 1	Low	-	
Setup website calendar		Working on it	! Jan 31 - Feb 24	Medium	-	
Link pages		Done	Feb 17 - 26	Low	-	
Connect firebase data to website		Done	Feb 5 - 23	Medium	-	
Make events addable to the calendar		Working on it	Feb 5 - Mar 1	Medium	-	
Remove username case sensitivity		Done	Feb 5 - 12	Low	-	
Develop UML Diagrams for system architecture		Done	Feb 20 - 26	Medium	-	
Cycle 1 Report		Done	Feb 15 - Mar 1	Medium	-	

Management Plan Visualized

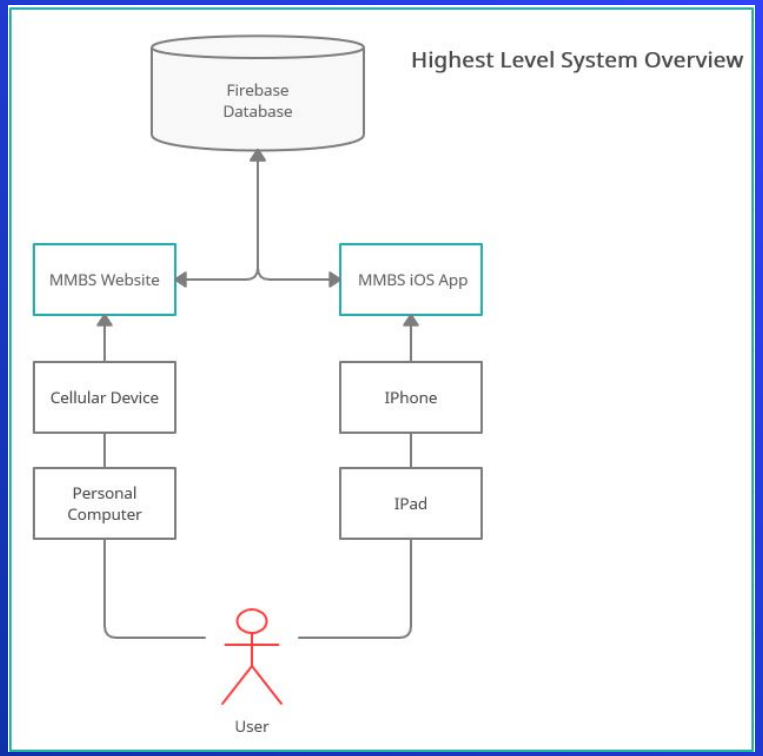
Design And Architecture





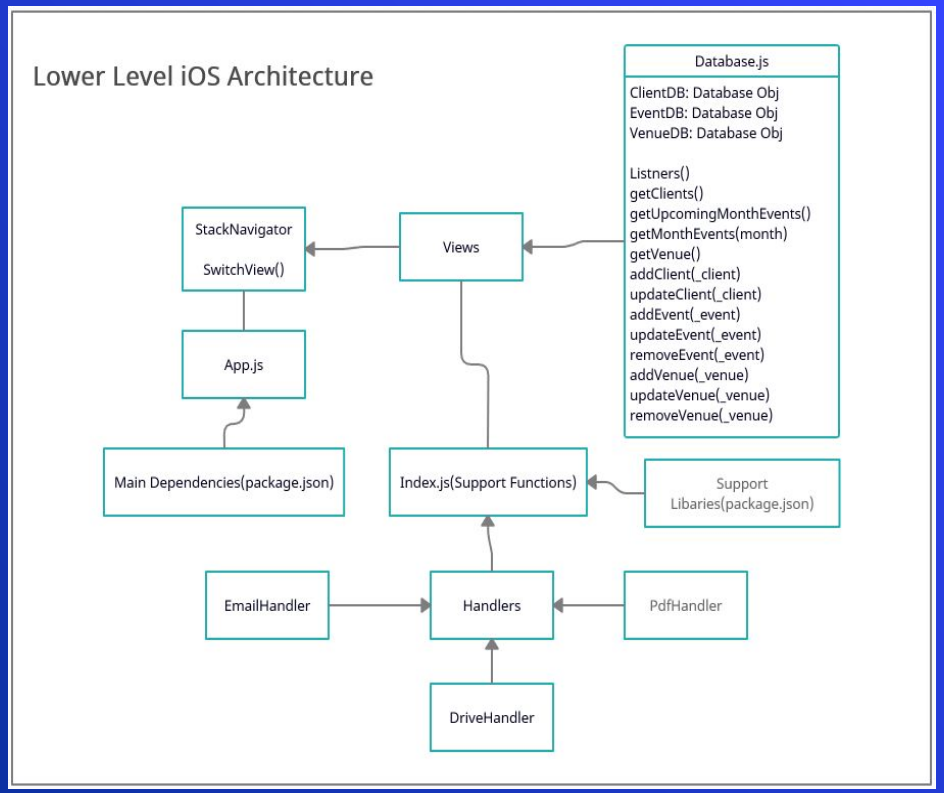
Credit : ReactNative.dev

High Level iOS Architecture



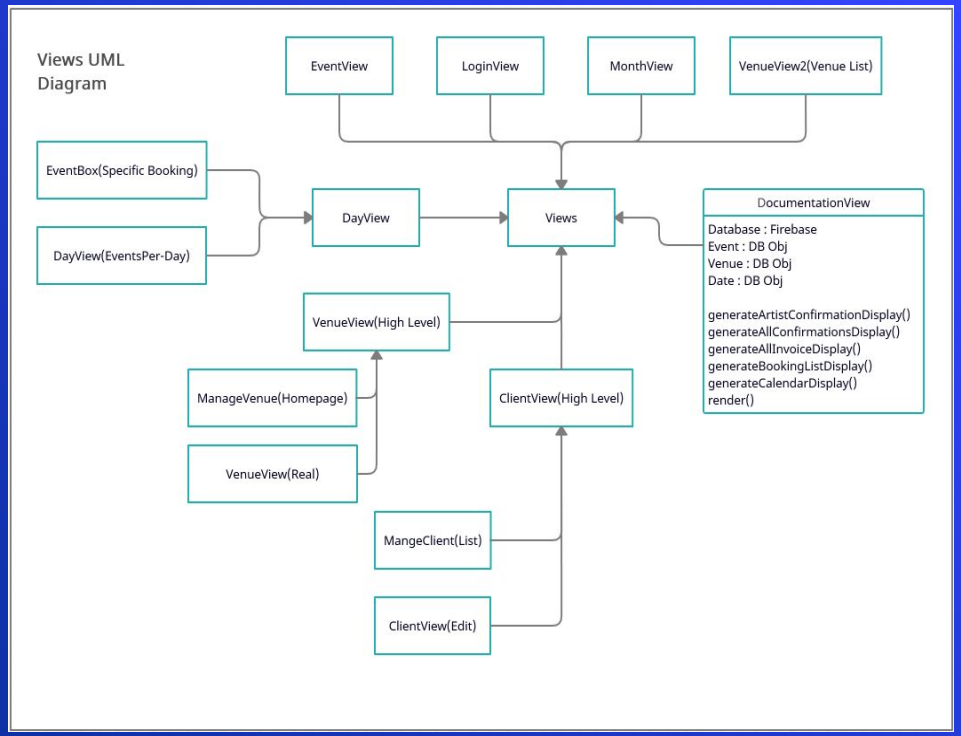
- Listener setup on both iOS and Website
- Updates application when data in database is changed
- Listener setup also on the database for any changes in the application
- Updates database on change

Low Level iOS Architecture



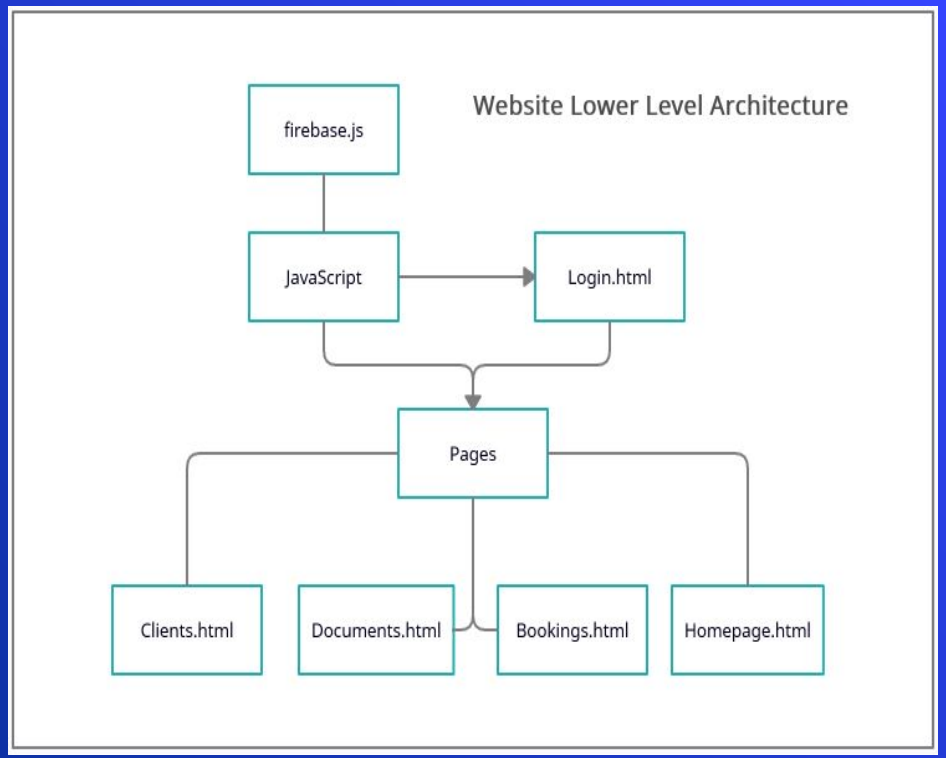
- App.js functions as main entry point
- Stacknavigator is setup from app.js and changes views.
- Views contains all iOS pages/screens
- Index.js functions as support for functionalities like emailer

Low Level iOS Architecture (cont)



- Views are what makeup most of the code and handle most the work.
- Certain views (due to the past teams) have multiple view classes.
- Documentation View has multiple variations and uses functions to display correctly.

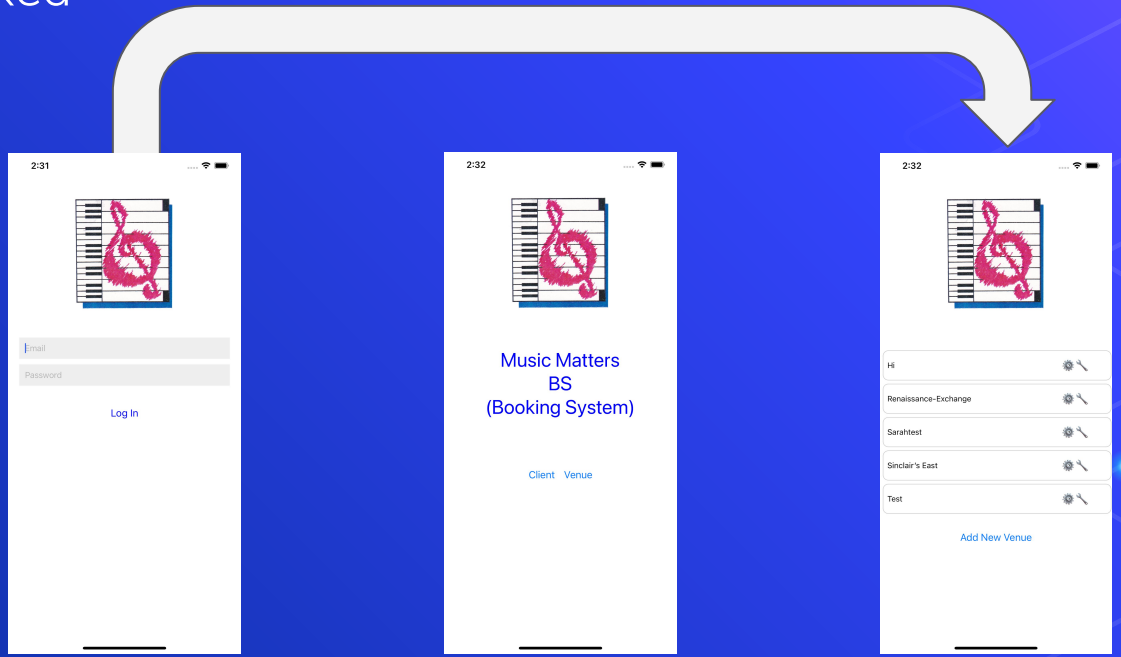
Low Level Web Architecture



- Login.html functions as the root directory and page of the website.
- All other pages are navigated to through Login.html.
- Javascript files accompany .html files to handle user interactions and support functions.

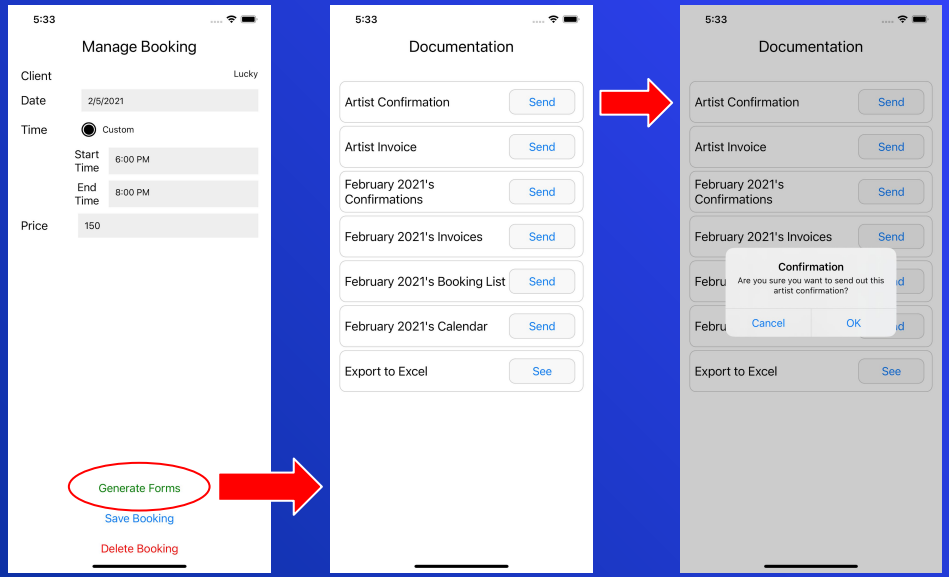
iOS User Story Progress

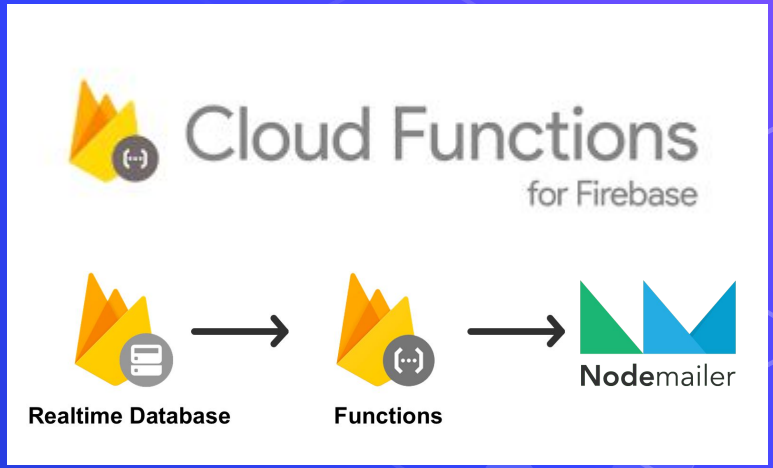
- Error Checking/Recovery
  - User Story: As a user, we want the homepage bug to be fixed



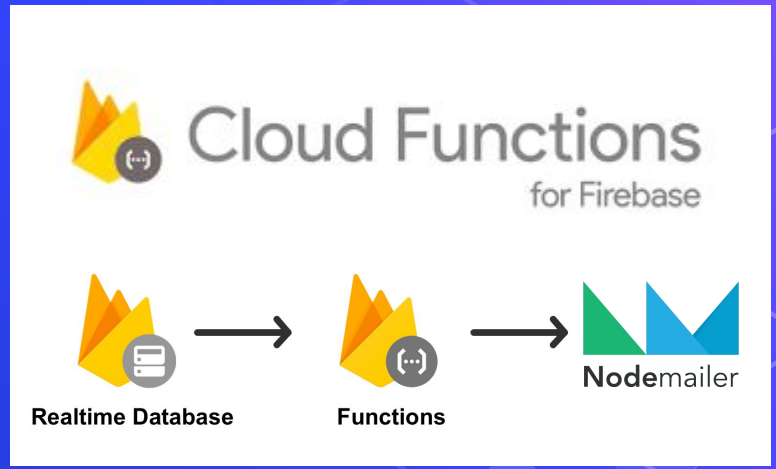
iOS User Story Progress (Cont)

- Functional Email System
  - User Story: As a user, we want to be able to choose client emails to be sent to the Music Matters Bookings email.





Products	Free Spark Plan Generous limits to get started	Pay as you go Blaze Plan Calculate pricing for apps at scale ✓ Free usage from Spark plan included*
A/B Testing	Free	
Analytics	Free	
App Distribution	Free	
App Indexing	Free	
Authentication		
Phone Auth - US, Canada, and India	10k/month	$0.01/verification
Phone Auth - All other countries	10k/month	$0.06/verification
Other Authentication services	✓	✓
Cloud Functions		
Invocations	*Not applicable*	Free up to 2M/month Then $0.40/million
GB-seconds		Free up to 400K/month Then Google Cloud pricing
CPU-seconds		Free up to 200K/month Then Google Cloud pricing
Outbound networking		Free up to 5GB/month Then $0.12/GB
Cloud Build minutes		Free up to 120min/day Then $0.003/min
Container storage		No free usage $0.026/GB
		*Pricing varies based on location
Cloud Messaging (FCM)	Free	
Crashlytics	Free	
Dynamic Links	Free	



Website User Story Progress

Website Dummy

Create a website that will be accessible through a users local computer and will be used as an example and starting point for the envisioned finished website.

Status: Completed

Cross-Platform Integration

If a user wishes to view the system calendar on their PC they can open a website or application, but if the user wishes to view the system calendar they can open it on their iOS device.

Status: Active Development

Website User Story Progress (Cont)

Ease of View

Create a calendar for the booking system on the PC application will be big enough to read clearly and will display artist names and performance times without having to click to expand.

Status: Active Development, Near completion

Fast and Clean User Experience

Get all areas of the app to perform as quick and responsive as possible, but also make them look appealing and user friendly.

Status: Active Development (will remain active until project completion)

New Website User Stories

E-mail functionality

Using the information from firebase, email the correct forms to performers about performances (or any other situation Mr. Moody might want to send an email regarding his company) automatically, or with a push of a button.

Status: Research, not started

Displaying real information on the calendar

Using the information from firebase and integrating it with FullCalendar properly, displaying the same (or perhaps more detailed) information compared to the iOS app.

Status: Early Active Development and research

Major Roadblocks

- Node.js was on the deprecated version 8
- Undocumented Code
- Repetitive Code
- Bugs within the app
- Code running out of order

Process

- Our process consists of a weekly cycle, one focused on communication. Our process has these steps:

- In simplified terms, we first communicate with our teammates about our planned additiontions during an individual/group coding session. These planned additions are ripped from our user stories.

- Once an individual session is completed, we record the time that session took personally, and then inform the teammates of what we accomplished. We usually talk to one another every few days, and we describe what we did and how things work during these calls if people don’t understand

Process

- Testing is usually followed up shortly after.
- Towards the end of the week, we tweak the UI of our additions to show the progress to Mr. Moody in our weekly meetings.
- During our meetings with Mr. Moody, we get his input on the app/website, and then he tells us what we should focus on and finish by next week.

Website and iOS Demo

Thanks!

Any questions?