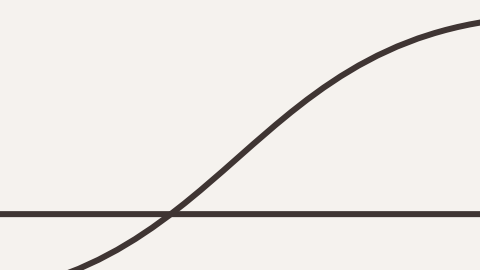




DASS Release 2

Booking app for Children Learning and Wellness centers



Team 49

Team Members

- | | |
|------------------|------------|
| - Divij | 2021101001 |
| - Nikunj Garg | 2021101021 |
| - Pranav Agrawal | 2021101052 |
| - Aman Raj | 2021101064 |

Problem Statement

- This app is intended for the parents to book child activities like origami, storytelling, art and craft, etc for their children.
- This app aims to make children wellness and learning centres easily accessible for the parents.
- App aims to help the franchise to manage their centre bookings, payment confirmations and accounting.

Motivation

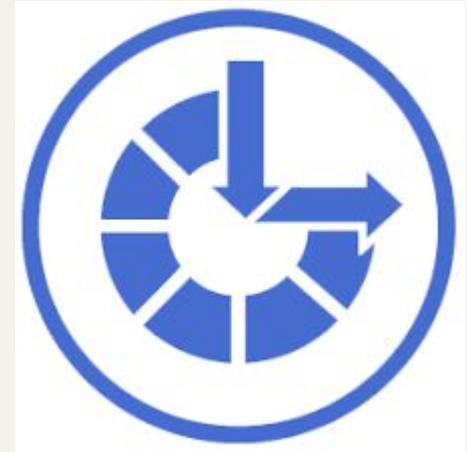
Main motivation behind the problem statement is to increase the ease of access to learning and wellness centers to parents. The app is required to be simple to achieve this. This helps both the parents by increasing their ease of access and the franchise by increasing their customer reach.



Increasing Ease of Access

How we plan to increase the ease of access and solve our problem statement?

- Make the app available on both Android and iOS
- Make the app intuitive and keep it simple.
- Show the centers based on proximity (as suggested by client) rather than availability.
- Add FAQs section and make it easily accessible to the user.



Increasing Customer Base

By increasing the ease of access to a user, we would be able to appeal to a large audience of parents who might not be that comfortable with technology or may be too busy to use a more time taking booking method like going to the center or making a call to the center.

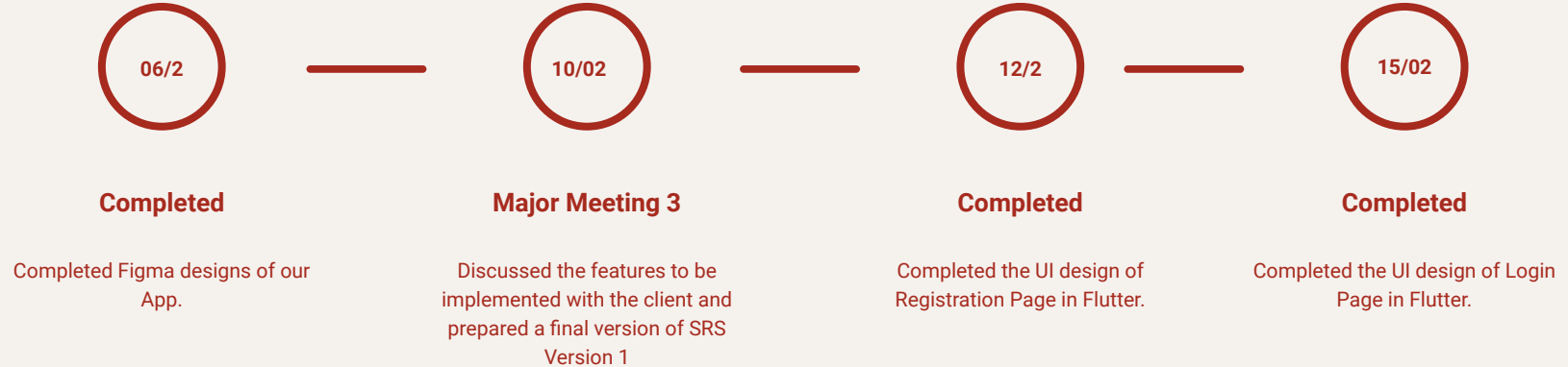
This means that more people now use the services given by learning and wellness centers hence, increasing their customer base.



Scoping



Scoping



Scoping



Completed

Created a UI design of the Profile Page with some basic features such as Add A Child, Edit Name and Phone Number.



Completed

Added Backend for the Registration Page and Login Page.



Completed

Added backend for the Profile Page.



Completed

Testing and debugging the Registration, Profile Page and Login Page. Added navigation between the pages as per UX decided.

Scoping



Completed

Team discussion about the Booking and Transaction module implementation.

Decisions made regarding schema of database(s) to use.



Completed

Created static pages for Activity and Party Booking without backend.



Completed

Implemented a static Location Page to find Nearest Locations to the user.



Completed

Added a static Payment Gateway (Razorpay) for the Transaction Module.

Scoping



Completed

Added backend for the Activity
Booking and Party Booking.



Completed

Testing and debugging of the
Booking and Transaction Module.



Completed

Added backend for the Location
Module. Tested and debugged the
Location Module.



Completed

Combined the Authentication
Module and Booking Module
together.

Scoping



Completed

Testing of the App and collection of user feedback from other people.



Completed

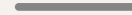
Redesigned some parts of User Interface based on the feedback from other people.



Completed

Team discussion about the UI/UX for the System Admin Module. Decisions made regarding tech stack to be used and schema of Database.

Started design of front end of System Admin Module



Major Meeting 4

Showed the client the App we designed and based on the feedback received, . revamped the UI/UX of the App.

Scoping



Completed

Completed the basic frontend of the System Admin Module.



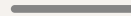
Completed

Completed the backend of the System Admin Module.



Completed

Integrated the front end and back end of the System Admin Module.



Completed

Testing and debugging of the System Admin Module.

Scoping



Completed

Completed the final testing of the User App.



Completed

Redesigned some parts in the User Interface of the System Admin Module.



Completed

Final testing of the System Admin Module was completed. .



Major Meeting 5

Shown the client the final version of our App. Client was satisfied with our App.

Tasks and Their Current Status

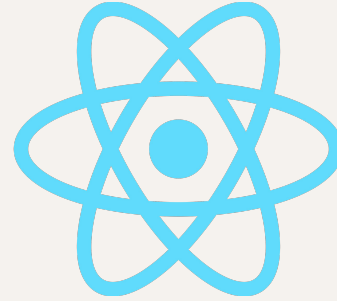
Task	Sprint Number	Status
User Profile : User Authentication	4	Completed
User Profile : Profile Page	4	Completed
Booking : Activity Booking	5	Completed
Booking : Party Booking	5	Completed

Payment : Billing Module	5	Completed
Payment : Transaction Module	5	Completed
System Admin Module : User Interface	6	Completed
High Level Design of App	3	Completed
System Admin Module : Database Management	6	Completed
Testing	7	Completed
My Bookings	8	Completed

Tech Stack Used

The following tech stack was used for the Project:

- **Flutter (version 1.0.0+1)**
- **RazorPay**
- **MongoDB**
- **Express.js**
- **React**
- **Node.js**



Flutter

- Flutter is a Software Development Kit (SDK) created by Google. It is used to develop cross-platform application for Android, iOS, Linux , macOS and Windows.
- The choice of Flutter for app development is supported by the large scale package library and community support available.
- Flutter is written in Dart programming language.
- It involves the use of:
 - Flutter engine (for low level rendering support)
 - Foundation Library (APIs for communicating Dart code with Flutter engine).
 - Design-specific Widgets (for creating Widgets)



MongoDB

- MongoDB is an open-source NoSQL database program.
- It uses JSON-like schema for storing data.
- It is written in C++, Python and JavaScript.
- MongoDB can be easily interfaced with Flutter by using the mongo_dart library (version : 0.9.0).
- Choice of mongoDB with it's community support.
- Requirement to integrate website of System Admin Module makes mongoDB a better choice over Firebase as MERN stack is very easy to work with.






- Razorpay provides payment gateway services to vendors, merchants, and e-commerce platforms.
- Razorpay also has a very robust Flutter plugin and provides easy integration.
- Since the app is an MVP, we are not required to do actual transactions. Doing a real transaction also requires information like GST IN which is unavailable to us. For such scenarios, Razorpay provides a test transaction mode.
- Razorpay provides Test API keys which can be used to simulate transaction without any actual money involved. These can be overviewed from their dashboard.
- Razorpay includes most of the famous transaction methods including external wallets.

Beyond the scope

We extended the scope of our project after completing all the major feature requirements. This was done in favour of increasing useability of the app, which we recognised as one of the most important quality goals of the app in our Software Architecture document.

We implemented the “My Bookings” functionality, where a user can see their previous bookings, marking upcoming and old bookings separately.

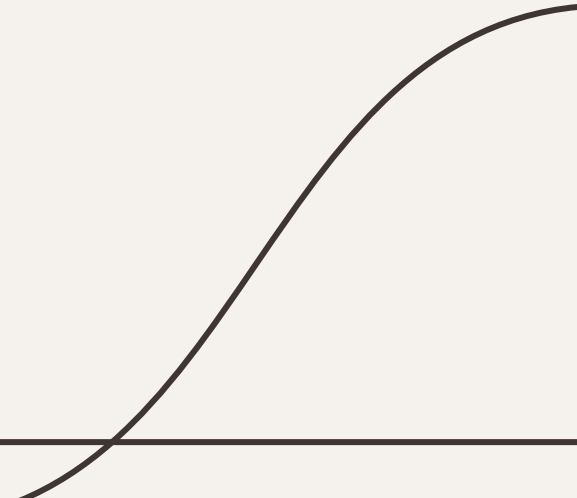
We also added small features to improve user experience and functionality like persistent login and invoice generation after successful payment.



Software Architecture Document

As requested by our client, we worked on **extra documentation**, to cover the needs of our project. Our Software Architecture document was built using a **industry standard arc-42 template** for the same.

Software Architecture document guides us to make architectural choices which are aligned to all the stakeholders. This was required because of the large scope of our project.



Thanks

