Home

Project Overview

DreamTeam is tasked with the design and development of a Personal Customer Relationship Manager in the form of a web application for a private client who runs a software business. Requirements include:

- Login and sign up
- Send emails
- Add, delete, update, label, and search contacts
- Schedule meetings with contacts and non-contacts
- Add, delete, update and search notes which can be associated with meetings
- Create, delete and update orders, which are associated with the contacts
- Update user profile
- Assign Labels to Contacts, and use labels to filter contacts

Navigate to …	Link
URL	https://dream-team-crm.herokuapp.com/
Git Repository	https://github.com/ccharansingh/DreamTeam
Current sprint-kanban board	Sprint 1 | Trello
Technology Stack	Technology Stack
Administrative Details	Administrative Details
Requirements	Requirements
Designs and Diagrams	Designs and Diagrams
Sprints	Sprints
Conventions	Style and Conventions
Testing Plan	Quality Assurance
Deployment Process	Deployment Process
Meeting Notes	Meeting notes

Meet The Team

Name (Student Number)	Role	Email	Interests
@ Anagha Giri	Product Owner	anaghag@student.unimelb.edu.au	
@ Chirag Charan Singh 1080617	Test Lead and Front End Dev	ccharansingh@student.unimelb.edu.au	Creating pages using REACT, Learning mocha, chai and other new ways of testing
@ kowji ,	Scrum master and FE/BE Dev	kowji@student.unimelb.edu.au	
@ Natasha Ireland 999132	Front End Lead	irelandn@student.unimleb.edu.au	Accessible design, learning REACT, utilizing Material-UI components
@ Olivia Ryan	Back End Lead	otryan@student.unimelb.edu.au	

AGILE Ceremonies

Ceremonies	Link
Stand ups	https://dreamteam56.atlassian.net/l/c/TXWJhuNU
Supervisor check in	Supervisor Check Ins

Retrospective	Retrospectives
Showcase	

Product Roadmap

Team mission

Roadmap overview

Administrative Details

Product Owner: Anagha

- Communication with clients
- Work with client to determine what features are required

Front-end Lead: Natasha

- Work with UX design team to design the front end of the product
- Present design to stakeholders and identify issues and concerns

Back-end Lead: Olivia

- Design and code for the back end of the product

Testing Lead: Chirag

- Organise and manage front and back end testing

Scrum Master: Koosha

- Check in with group members to check we are on track and keeping to schedule
- Manage obstacle clearing
- Reallocate resources

Client Contact Info

ID: 1049166

Name: Wei, Yiran

Email: yirwei@student.unimelb.edu.au

Group/Working Conventions

Formal Decision Making Process

Case: task owner comes forward with decision

- options in decision are thought out and documented
- impact of decisions is clearly outlined
- supporting evidence for options is provided where necessary

Solution: team who has taken ownership of this development area (e.g. front end, back end, testing etc) takes responsibility of this decision

- all members of the responsible party must agree on a decision
- where no agreement is reached within at most one discussion (lasting no longer than 15 minutes) the decision is taken to the scrum master and cases made for all options

Requirements

This folder contains all the requirements for the project DreamCRM, created via meetings with the client.

Navigate to	Link
User Stories	User Stories

Motivational Model	Motivational Model
User Persona	User Persona
Use Cases	Use Cases

User Persona

Yiren

Job Title
Student

Age
18 to 24 years

Highest Level of Education
Master's degree (e.g. MA, MS,

Social Networks

Industry
Technology

Organization Size
Self-employed

Preferred Method of Communication

- Email
- Zoom Meetings

Tools They Need to Do Their Job

- CRM Software
- Email

Their Job Is Measured By

Productivity

Goals or Objectives

Categorize contacts based on tags, locations, company, and role.
Store all contact information, history and communication for instant access.
Organize schedules to keep track of events and meetings.
Have unified access to communication channels like email
See all important details and past engagements when dealing with a contact.

Biggest Challenges

- Navigating Client Relationships & Communications
- Communication
- Collaboration & Creativity

Use Cases

1. User can create an account by entering their name and email and creating a username (default) and password
2. User can login by entering their email/username and password
3. Add a contact: name, contact details, label, etc
4. Create a label: colour, name
5. Edit an existing contact: edit name, edit/delete label, delete contact
6. Add a note
7. Delete a note
8. Schedule a meeting
9. Log a meeting: enter details such as time and date, attendees (link other user accounts or add names of non-users), notes
10. Send email to a contact through the CRM UI.

User Stories

The user in the table below refers to our client Yiran, who will be the only one using the system at this stage.

The priorities were determined primarily from meetings with the clients and their preferences.

ID	As a <role>	I want to <do something>	So that <achieve some goals>	Priority

1	new user	create an account	I can create an account to access the CRM system	HIGH
2	user	can log in using email and password	I can securely access the CRM system	HIGH
3	user	can schedule meetings	I can view all the details about my past and upcoming meetings in one section	HIGH
4	user	can create new contacts	I can easily view their information and schedule meetings with them	HIGH
5	user	can edit or delete existing contacts	I can update any new contact information, such as labels or contact emails	HIGH
6	user	can label relationships with other contacts	the contacts can be easily sorted on the basis of their label	HIGH
7	user	can view and edit any upcoming meetings	I can check my schedule and update it accordingly	HIGH
8	user	can create, edit and delete notes for meetings	I can easily take notes during the meetings to ensure no information is lost	HIGH
9	user	can email other contacts	I can send direct messages to the clients/colleagues	HIGH
10	user	can create orders	I can store all the required information regarding the outgoing orders of the business	MEDIUM
11	user	can edit, update or delete orders	I can change any information regarding the orders as it changes	MEDIUM
12	user	can edit profile information	I can change my details if that every changes	MEDIUM
13	user	can visualise data from my previous orders	I can easily view and collect data from my previous orders to guide me towards making relative decisions	VERY LOW

Updated requirements following sprint 1 review with client:

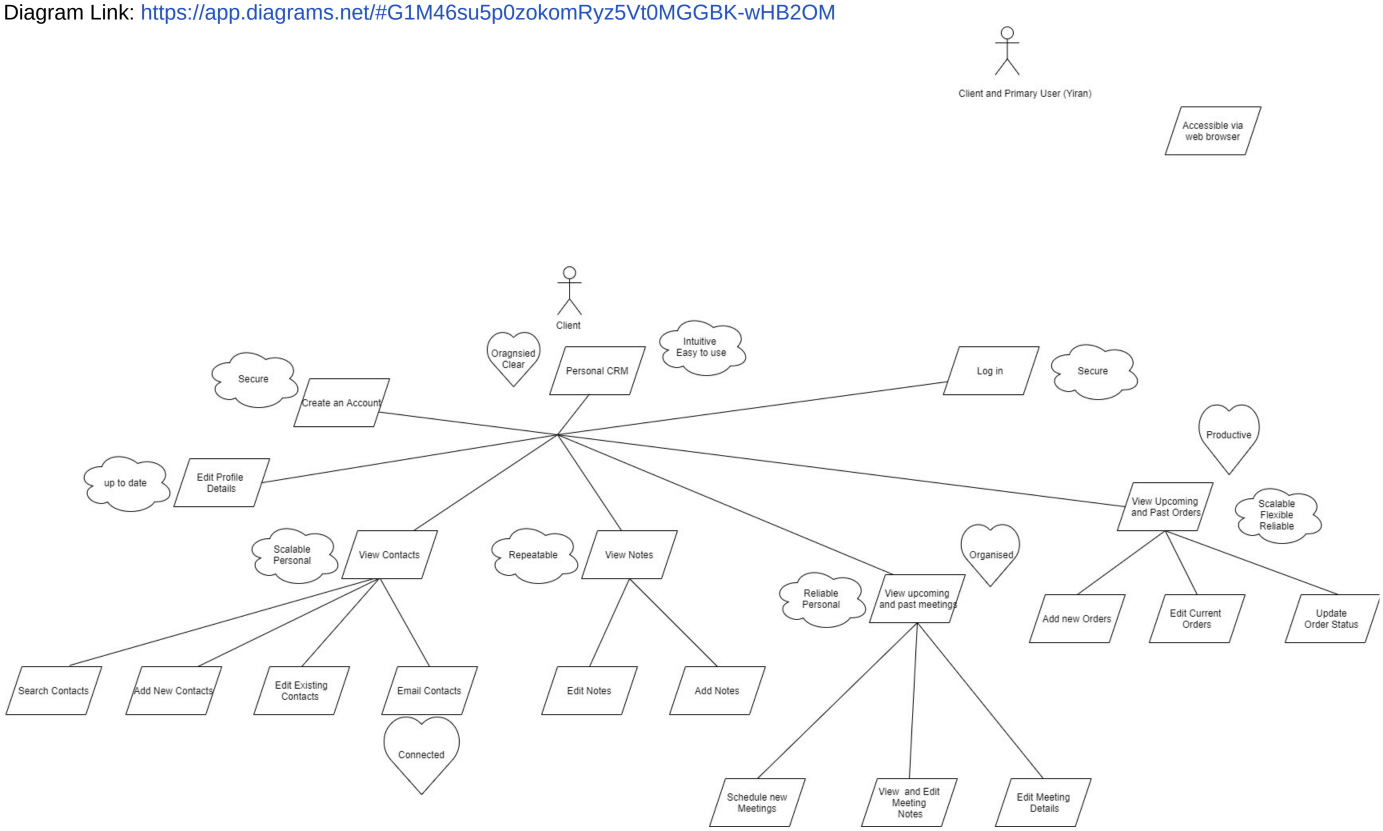
ID	As a <role>	I want to <do something>	So that <achieve some goals>	Priority
14	user	filter notes	I can search for notes based on title	HIGH
15	user	search contacts	I can filter contacts by searching for their first name	LOW
16	user	order notes	I can order notes by time or title (alphabetical)	LOW

Motivational Model

Who - Stakeholders	Do	Be	Feel	Concerns
Client (Yiran)	create an account	reliable	organised	
	log in	secure	satisfied	
	view upcoming and past Meetings	flexible	productive	
	edit meetings	Intuitive	clear	
	view and edit meeting notes	Repeatable		
	schedule new meetings	Personal		
	view contacts	scalable		

	create new contacts	easy to use		
	search contacts	up to date		
	edit contacts			
	email contacts			
	add contacts			
	view notes			
	edit notes			
	add notes			
	view current and past orders			
	add new orders			
	add order details			
	update order status			
	edit Account Details			

Diagram Link: https://app.diagrams.net/#G1M46su5p0zokomRyz5Vt0MGGBK-wHB2OM



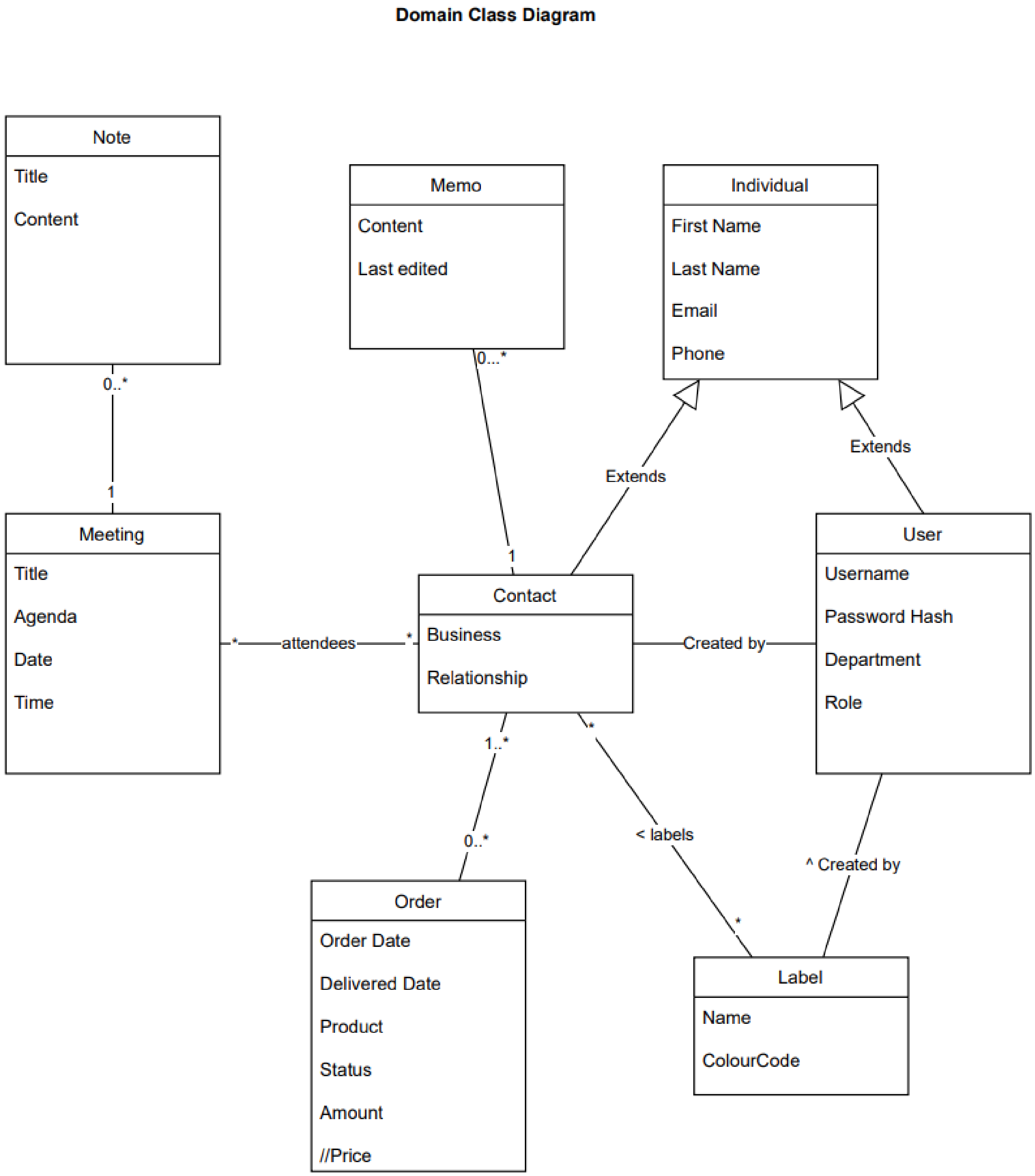
Designs and Diagrams

This folder contains the details and iterations of all the designs created during the Inception Sprint of DreamCRM project.

Navigate to	Link
Domain Model	Domain Model
Design Class Diagram	Design Class Diagram
Database Model	Database Model
UI design	UI Design
Colour Theme	Colour Theme

Domain Model

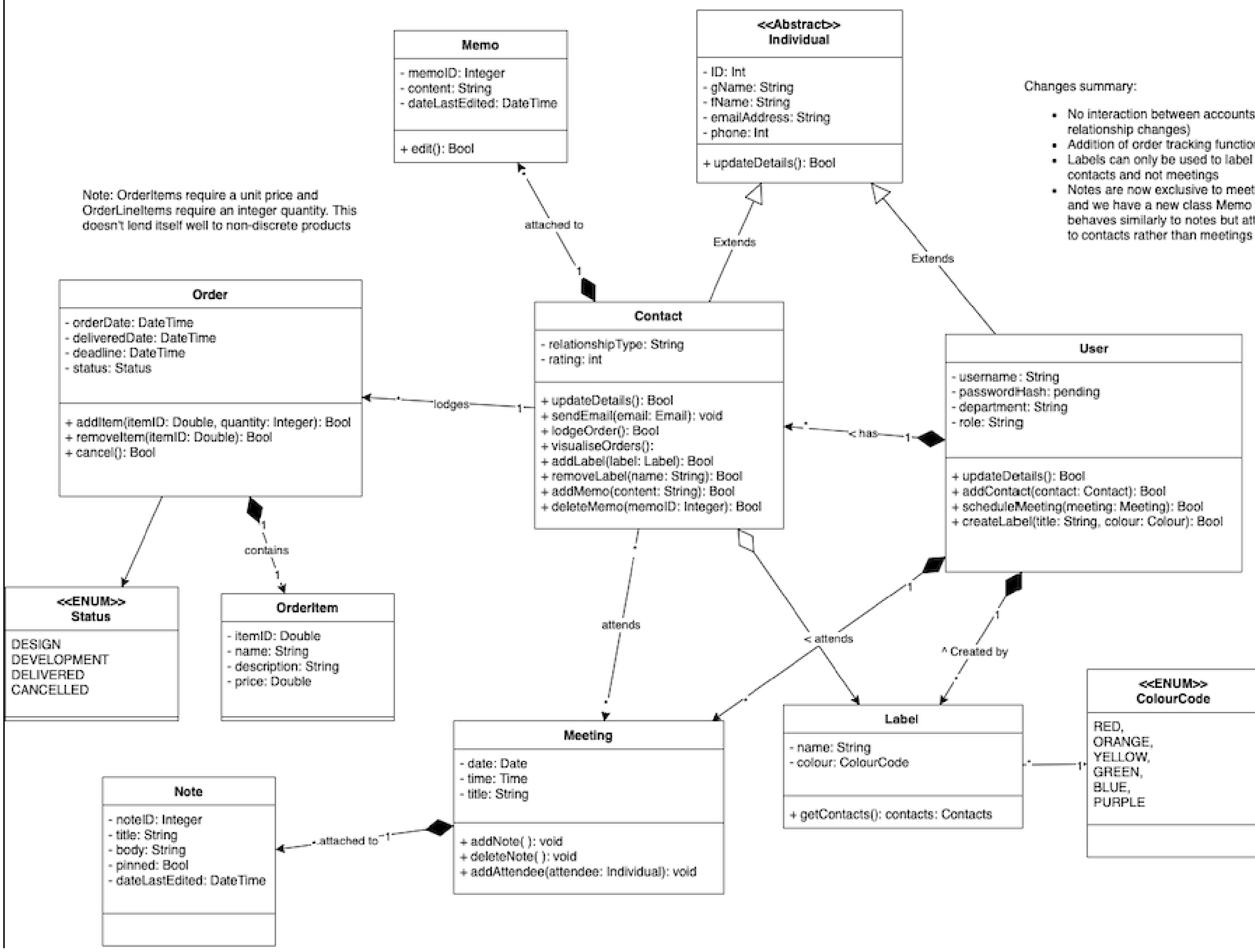
Domain Model - amended after feedback from supervisor



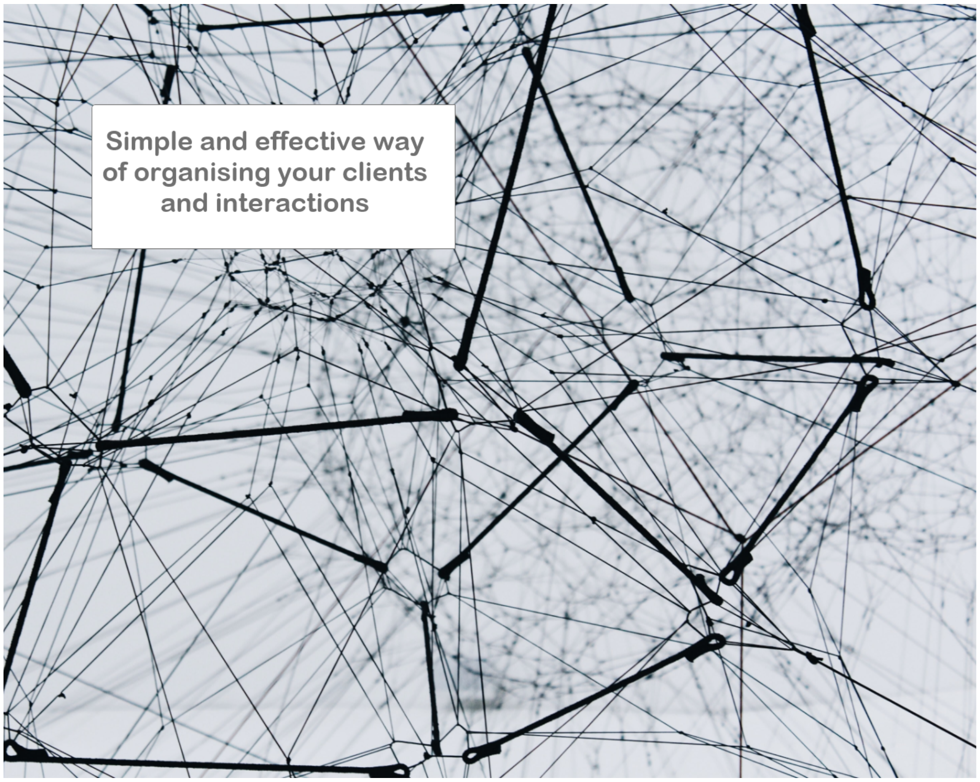
https://app.diagrams.net/#G1Hq0uCHUtTrt7GHWXTwYdinHy9TnFsXim

Design Class Diagram

Design Class Diagram



UI Design



Let us (net)work for you

Email

Password

LOGIN

Don't have an account?

SIGN UP

Customer Relationship Management brought to you by

The Dream Team PTY

Intellectual Property of the University of Melbourne

< Back

Thank you for joining the Dream Team

Create An Account

Given Name

Family Name

Email

Password

Confirm Password

Phone

Department/Role etc

The Dream Team PTY

Your password must contain atleast one uppercase letter, one numerical character and one special character

SIGN UP

Back

Edit Profile

Given Name

Family Name

New Password

Confirm New Password

Department, Roles etc

Your password must contain atleast one uppercase letter, one numerical character and one special character

Update

The Dream Team PTY

Logout

Home

Meeting

Schedule a meeting with a contact
Check your upcoming meeting
Prepare for your meetings

Contacts

Add a contact
Remove or update contacts
Label your contacts

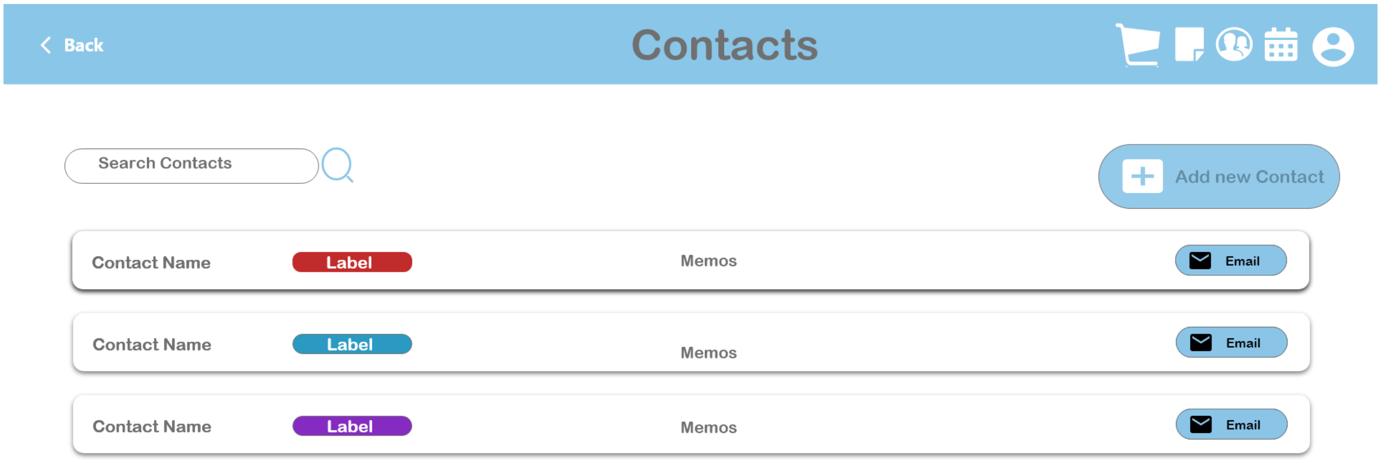
Orders

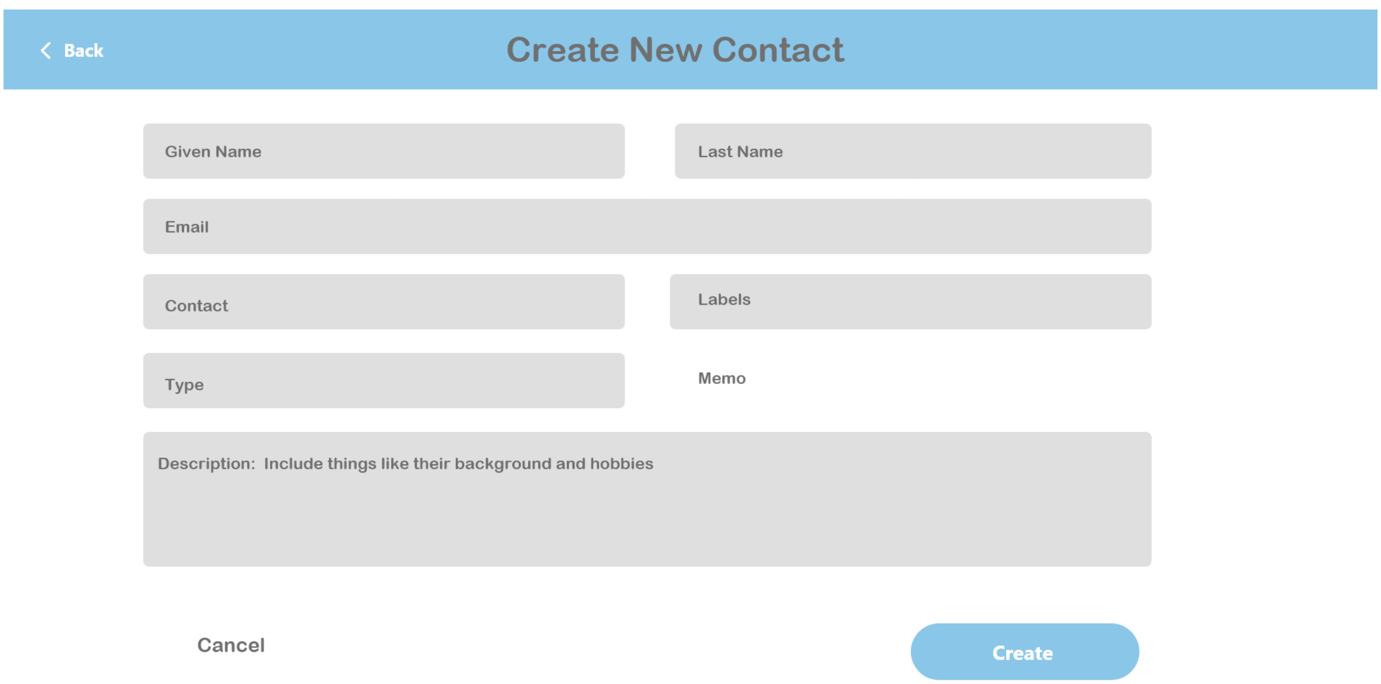
Create your orders
Check progress of orders
Edit orders

Notes

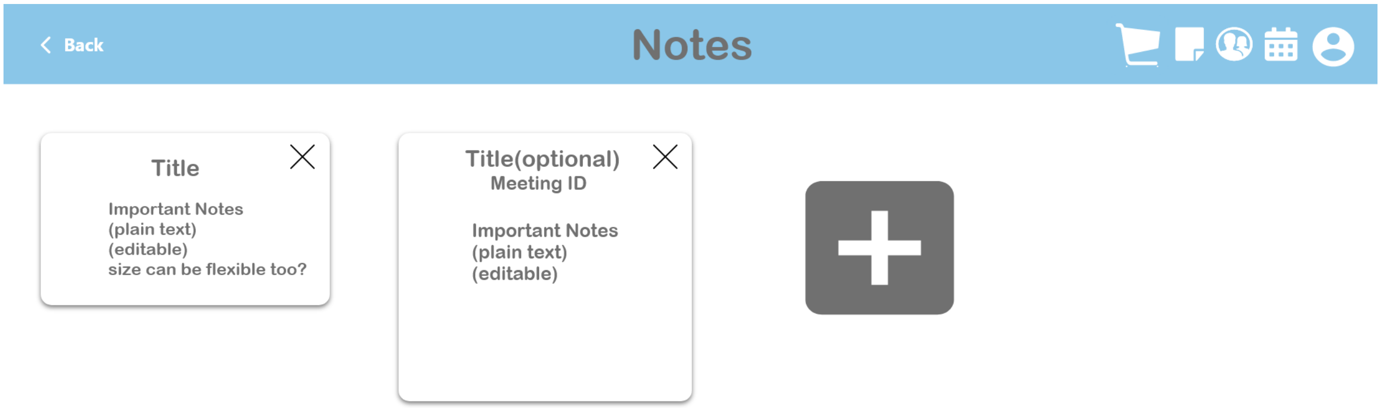
Write a quick note
Read past notes
View/Edit Notes from Meetings

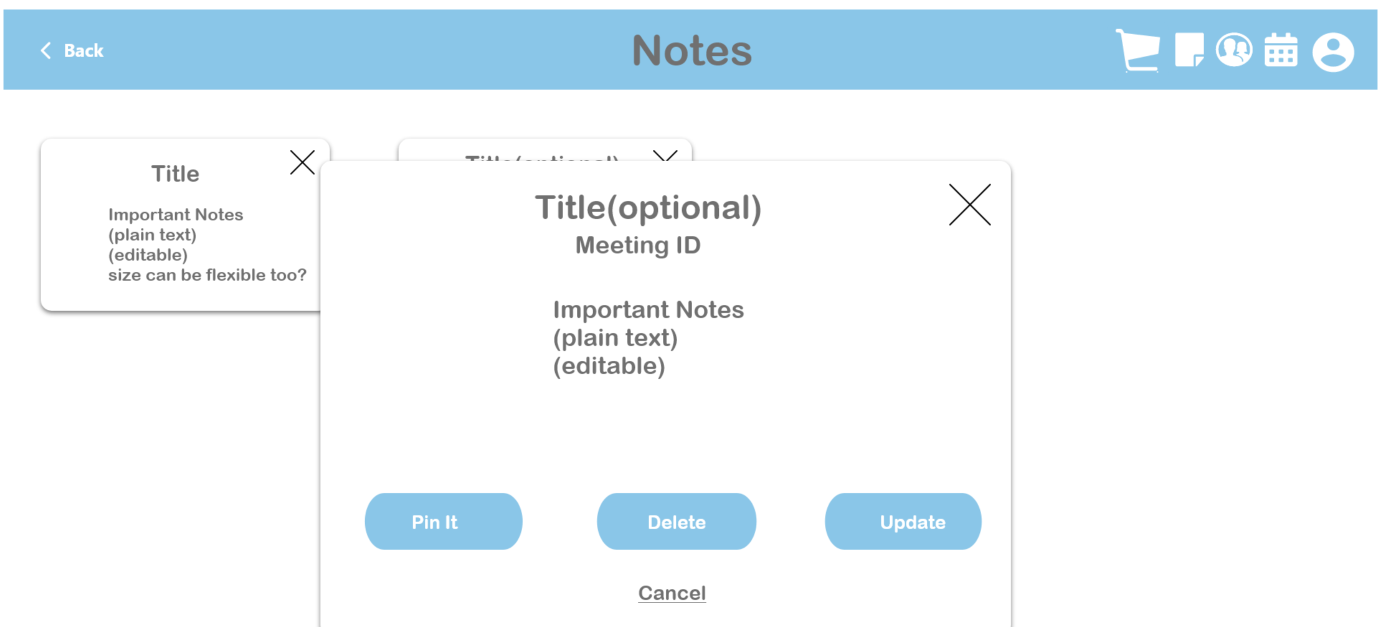
The Dream Team PTY

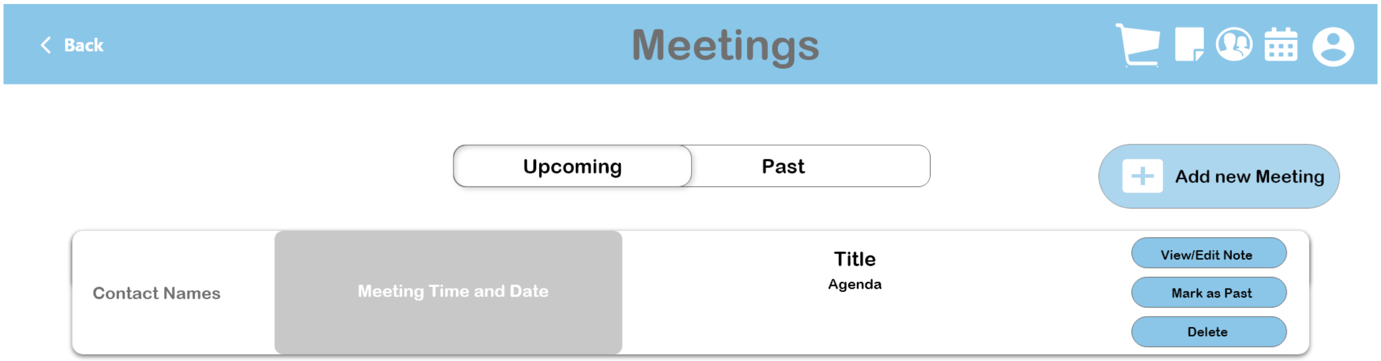


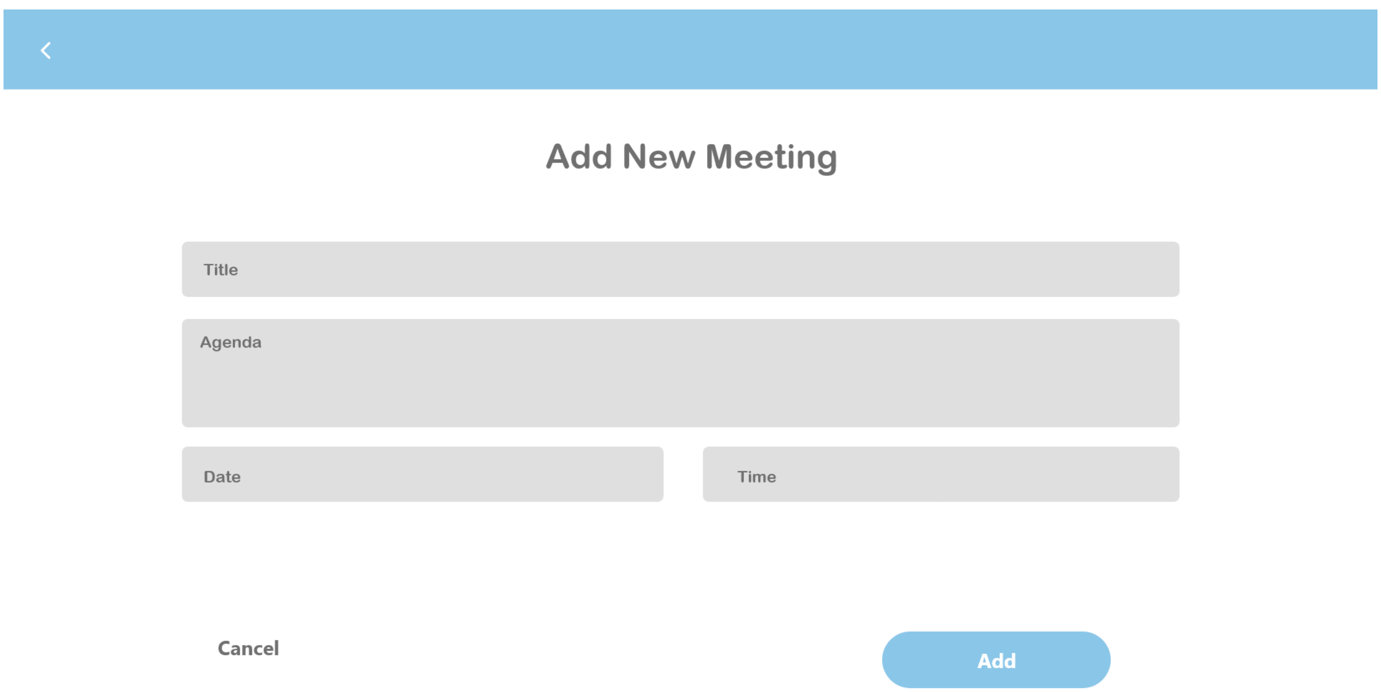


The Dream Team PTY

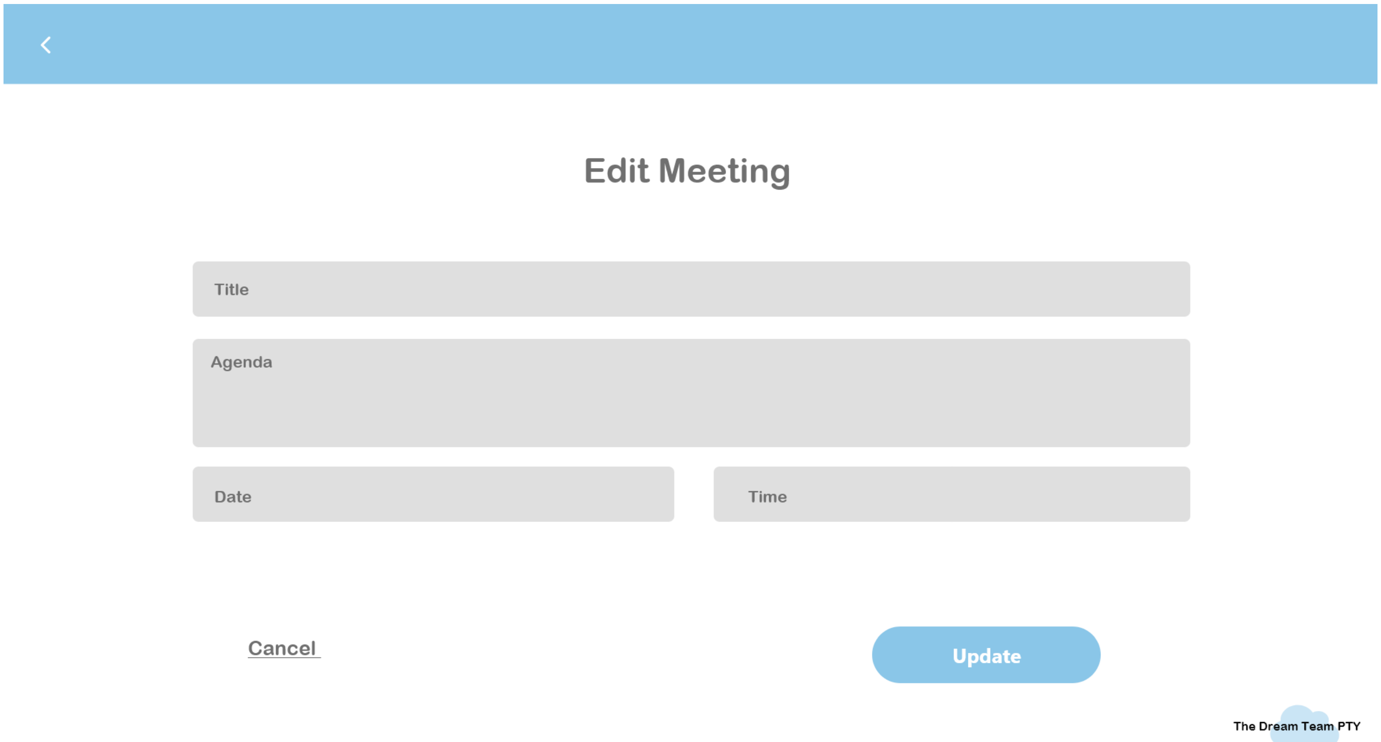

Back
Notes
Title
Important Notes
(plain text)
(editable)
size can be flexible too?
Title(optional)
Meeting ID
Important Notes
(plain text)
(editable)

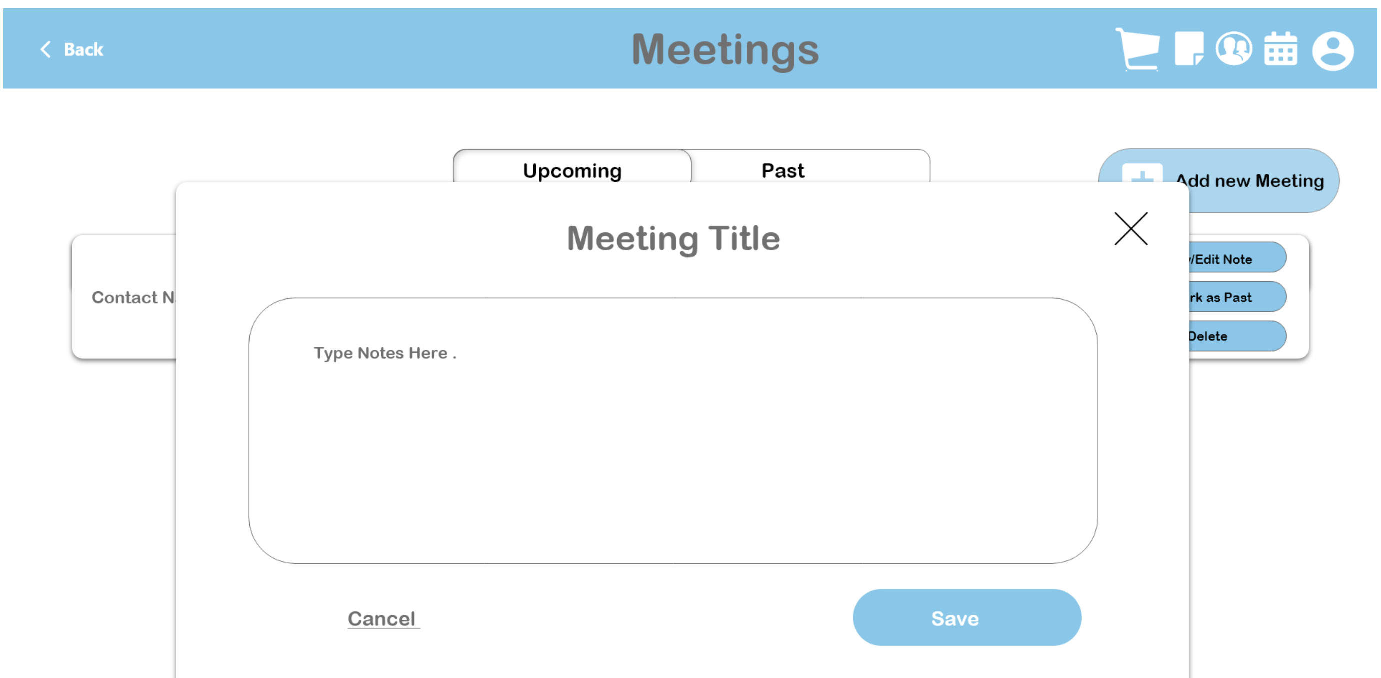

Back
Notes
Title
Important Notes
(plain text)
(editable)
size can be flexible too?
Title(optional)
Meeting ID
Important Notes
(plain text)
(editable)
Pin It
Delete
Update
Cancel





The Dream Team PTY


Edit Meeting
Title
Agenda
Date
Time
Cancel
Update
The Dream Team PTY


Back
Meetings
Upcoming
Past
Add new Meeting
Contact N
/Edit Note
rk as Past
Delete
Meeting Title
Type Notes Here .
Cancel
Save

Back

Orders

Add new Order

Delivered Orders

Order ID	Status/Progress	Design	Actions
342	Item	whatever she will sell/buy	Update
	Quantity	2	
	Tracking	trackithere.com.au	Delievered Delete

Order ID	Status/Progress	Development	Actions
078	Item	socks	Update
	Quantity	a few	
	Tracking	0414040404	Delievered Delete

Client ID	Status/Progress	Deployment	Actions
342	Item	CRm	Update
	Quantity	2	
	Tracking	0404040404	Delievered Delete

Back

Create New Order

Item

Quantity

Contact Details

Status

Lodged Date

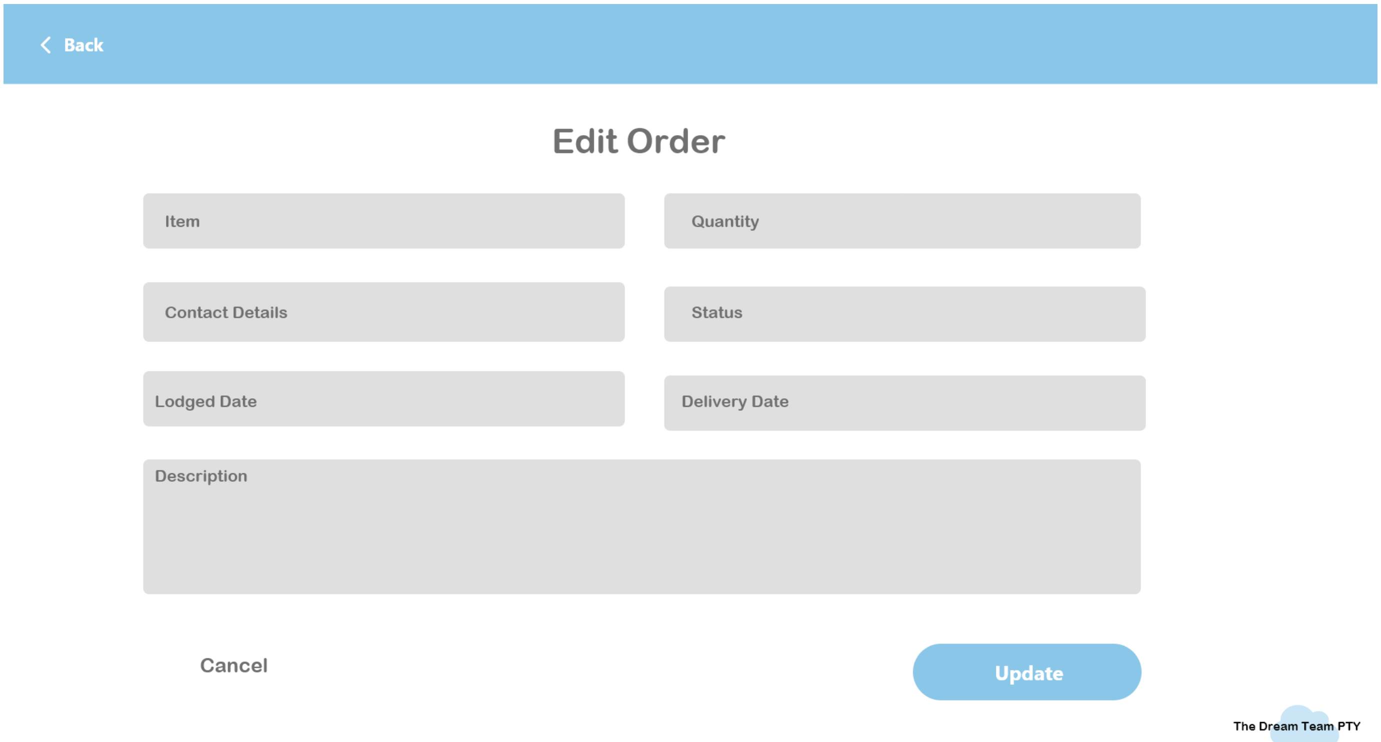
Delivery Date

Description

Cancel

Create

The Dream Team PTY



These designs are subject to modifications to improve usability.

Colour Theme

Material UI Accessibility Analysis Tool: Color Tool - Material Design

Primary

#5a6abf

	Aa Large Text	Aa Normal Text
White Text	min 63% opacity	min 93% opacity
Black Text	min 65% opacity	NOT LEGIBLE

P — Light

#8d98f2

	Aa Large Text	Aa Normal Text
White Text	NOT LEGIBLE	NOT LEGIBLE
Black Text	min 49% opacity	min 66% opacity

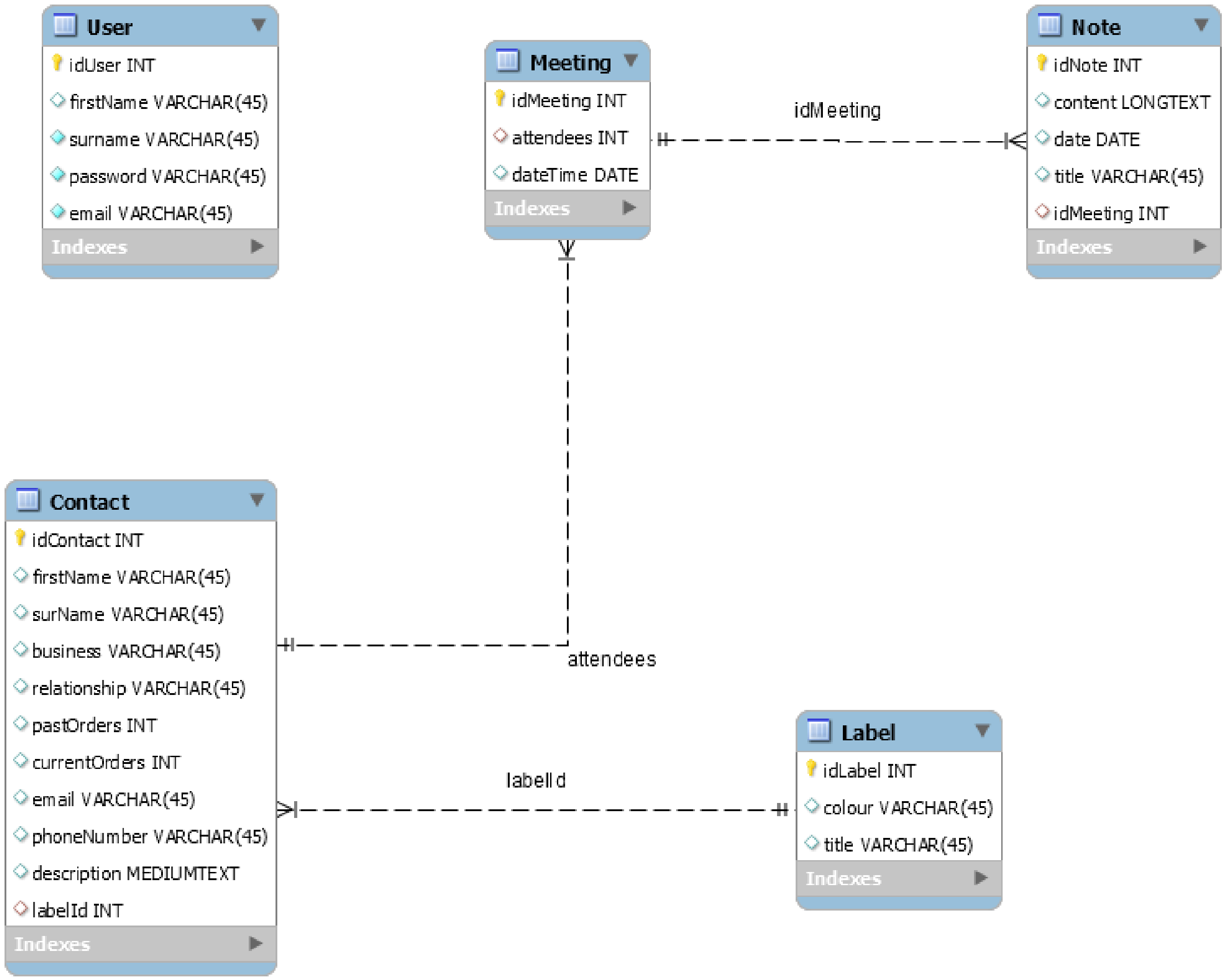
P — Dark

#23408e

	Aa Large Text	Aa Normal Text
White Text	min 42% opacity	min 60% opacity
Black Text	NOT LEGIBLE	NOT LEGIBLE

Database Model

Draft in sql just to give a rough overview. We will be using mongoDB so noSQL but can see types and attributes as shown.



Still iterating over what to do with notes.

Technology Stack

Front End	• React
Back End	• MongoDB • Nodejs
Testing	• Unit Testing (Mocha) • Integration Testing(Mocha, chai) • User Acceptance Testing
API Documentation	• Postman
Deployment	• Github Actions (pipeline deployment) • Heroku

Sprints

This folder contains all the relevant documents to the Sprints. Sprint Planing Process contains the process used to plan the sprints.

- **Inception Sprint**

Navigate to	Link
Sprint Report	Inception Sprint Report

Retrospective	Retrospective - Inception Sprint

- **Development Sprint 1**

Navigate to	Link
Sprint Planning	Sprint Planning: Development Sprint 1
Sprint Review	Development Sprint 1 - Sprint Review
Sprint Report	Development Sprint 1 Report
Retrospective	Retrospective - Development Sprint 1
Testing Report	Testing Report - Development Sprint 1

- **Development Sprint 2**

Navigate to	Link
Sprint Planning	Sprint Planning: Development Sprint 2
Sprint Review	Development Sprint 2 - Sprint Review
Sprint Report	Development Sprint 2 - Sprint Report
Retrospective	Retrospective-development sprint 2
Testing Report	Testing Report - Development Sprint 2

- **Development Sprint 3**

Navigate to	Link
Sprint Planning	Sprint Planning: Development Sprint 3
Sprint Review	TBC
Sprint Report	TBC
Retrospective	TBC
Testing Report	Testing Report - Development Sprint 3

Inception Sprint

Navigate to	Link
Sprint Report	Inception Sprint Report
Retrospective	Retrospective - Inception Sprint

Retrospective - Inception Sprint

Date	09 Sep 2021
Team	Dream
Participants	@ Natasha Ireland @ Olivia Ryan @ Anagha Giri @ kowji @ Chirag Charan Singh

Background

This retrospective corresponds to the Inception Sprint of Dream CRM project. During this sprint, the client was consulted regarding their requirements from the system. Those requirements were then used to construct user stories, domain models, UI designs and Motivation Models. In addition to this, the specifications and administrative details were laid out during this Sprint.

Retrospective

Start doing	Stop doing	Keep doing
• Have more clear and detailed meeting notes and record meetings from stand-ups, supervisor meetings, and meetings with the client • Categorise tasks based on priorities within the sprint • Dividing the development work within the team • Better usage of our Trello space • Link to external designs and internal pages when references are made on the confluence • Categorise and Label tasks based on the time and resources required	• Keep changing decisions about how notes will be connected to meetings • Communicating on messenger regarding Project decisions	• Asking lots and lots of questions • Discussing any conflicts between models as a team to work out the best way to proceed • Reviewing and updating models to match recent changes to design • Taking the time to help others, especially ensuring a concrete understanding when things go wrong

Action items

- [ ] Prioritise using Trello board to log, assign and update tasks including linking progress where possible
- [ ] Adopt a systematic approach in terms of assigning tasks.
- [ ] Enabling extensions of the base design (as outlined by the client) to be discussed and decided on as a team so that individuals aren’t left to make large design decisions that will impact other’s work alone

Inception Sprint Report

Dates	26/7/21-16/8/21 (rough)

Contributions by team member	**<u>Chirag</u>** Motivational model Personas Sprint 1 user stories final Front end adobe XD draft 1 **<u>Koosha</u>** Set up planning poker DB schemas draft 2 **<u>Tash</u>** Sprint 1 user stories draft Front end adobe XD draft 1 DB schemas draft 2 **<u>Olivia</u>** Sprint 1 user stories draft and final Domain class diagram Product roadmap draft **<u>Anagha</u>** Client communication **<u>Team</u>** CRM research Role allocation Sprint 1 planning and story point allocation Tool selection
Sprint achievements	Roles allocated: https://dreamteam56.atlassian.net/l/c/uqBPiSHd Roadmap: Product Roadmap Tool selection: Technology Stack Git repo setup: ccharansingh/DreamTeam: Personal CRM for IT Capstone Project (github.com) Client requirements initial: Use Cases User stories: User Stories DB schemas: Database Models/Schemas Domain class diagram:Design Models
Backlog items, status	Motivational model-awaiting allocation
Backlog items from previous sprint	NA
Backlog items removed from sprint	NA
Link to kanban board	
Link to retrospective	Retrospective - Inception Sprint

Development Sprint 1

Navigate to	Link
Sprint Planning	Sprint Planning: Development Sprint 1
Sprint Review	Development Sprint 1 - Sprint Review
Sprint Report	Development Sprint 1 Report
Retrospective	Retrospective - Development Sprint 1

Sprint Planning: Development Sprint 1

User story: login and sign up

BE: L

FE:

1. Set up app.js (small)
2. Finish schemas (medium)
3. Set up the database online with schemas from design (medium)
4. Connect to the database (medium)
5. Define user model (small)
6. Implement Bhash for hashing the password to store in db (medium)
7. Implement password rules (medium)
8. Set up login/signup token validation (use passport.js) (large)
9. Set up sign up page (VMC)- front design (medium)
10. Set up login page (VMC) -front end design (medium)
11. Test login (large)
12. Test sign up (large)

User story: user can send email to contact (email functionality will be API dependent- to be looked at later when more information is available)

BE: L

FE:

1. Research email API
2. Identify email API of choice to use
3. Back end integration of email API with app (may be broken down further)
4. Front end integration of email API with app (may be broken down further)
5. Testing

** More steps likely- sprint update required**

User Story: user can create and delete labels, assign colour

BE: M

FE:

1. Set up label schema in mongo
2. Define label schema in js
3. Connect to label collection
4. Controller for add
5. Controller for delete
6. Controller for assign colour (findByIdAndUpdate ?)
7. Routes for add, delete, assign colour
8. Front end design for these functions
9. Unit test

User Story: user can add, delete contacts

BE: L

FE:

1. Set up contact schema in mongo (medium)

2. Define contact model in js (small)
3. Connect to db collection contact (small)
4. Controller for add (medium)
5. Controller for delete (medium)
6. Routes for add and delete (medium)
7. Front end design for these functions (large)
8. Unit tests for route, controllers (large)
9. Integration tests for route, controllers (extra large)

contact memos will be stored as an array of json files that contain the content AND the date posted in the contact model, need ID

Sub User Story: User can add or delete label, assign memos to contact

BE: S

FE:

a) once user has selected a contact they can choose to assign an existing label or create a new one and assign that

1. Controller for add label to contact
2. Controller for delete label on contact
3. Controller for add memo
4. Controller for update memo
5. Controller for delete memo
6. Route for add and delete
7. Routes for memo functions
8. Front end design for each function (on contact page)
9. Test controllers
10. Test routes

User Story: User can create, delete and update notes

BE: L

FE:

1. Set up note schema in mongo
2. Define note model in js
3. Connect to note collection
4. Controller for add
5. Controller for delete
6. Controller for update
7. Routes for add, delete, update
8. Front end design for each function
9. Integration test
10. Unit test

User Story: user can schedule and cancel meetings

BE: L

FE:

1. Set up meeting schema in mongo
2. Define meeting model in js
3. Connect to meeting collection
4. Controller for schedule
5. Controller for cancel
6. Routes for schedule, cancel
7. Front end design for each function
8. Integration test
9. Unit test

User Story: user can create, update and delete notes from a meeting

BE: L

FE:

1. Controller to create note from meeting
2. Controller to update
3. Controller to delete
4. Route for create, update, delete
5. Front end design for this
6. Unit test
7. Integration test

(tentative note functionality: have generic notes page and a link on meeting to view all notes associated with that meeting: would be redirected to localhost:3010/notes/meetingID=token etc

Retrospective - Development Sprint 1

Date	09 Sep 2021
Team	Dream
Participants	@ Natasha Ireland @ Olivia Ryan @ Anagha Giri @ kowji @ Chirag Charan Singh

Background

The primary goal of this sprint was to produce fully functional login, sign up, notes and contacts pages. In addition to this, the sprint comprised of setting up authentication, database models and all the related dependencies.

Retrospective

Start doing	Stop doing	Keep doing
• Merging completed changes to main • Back any decision making with pros and cons • Increase backend-frontend communication • Updating tasks on trello as they are started or completed • More regular pushes to remote repo • Testing	• Non-standard git-branch naming • Unconventional code format: i.e. backend is mostly snake case and front end is camelcase, middleware is camelcase. Also indentation inconsistencies • Not commenting code	• Pair programming • Presenting problems to the team • Being happy to jump on someone's branch to help troubleshoot • Skill and knowledge sharing

Action items

- [x] Create code convention confluence page (branch names, merging habits, code style, error-handling etc)
- [x] Keep track of features for testing purposes
- [x] Comment code
- [x] Convention for error catching
- [ ] Create a generic error page to return when functions run into errors
- [x] Push to remote at least once every four commits

Development Sprint 1 - Sprint Review

Background:

Date: 10/09/2021 at 11:30 am

Attendees: The Dream Team (Anagha, Chirag, Koosha, Natasha and Olivia) and Client (Yiran)

Agenda: Show the client the progress since our previous meeting and to get feedback on the deliverables created during the sprint.

Meeting Minutes:

Refer to link below for in-depth meeting minutes and the recording of meeting.

Client Meetings

Client Response/Feedback:

Commendations:

- Likes the confirmation pop-up when contact is being deleted

- Likes the layout for the meetings page (to be elaborated and completed in the next sprint)
- Likes the colour and format of the website - even if the colour is a bit different to original UI design shown

Recommendations:

- Would like to make first name and email mandatory for contacts
- Would like a filter search bar
    - Notes: High Priority - search by title
    - Contact: Low Priority - search by name
- Would like to be able to order notes by either time or title (alphabetical) Low Priority

Development Sprint 1 Report

Dates	16/8/21-6/9/21
Contributions by team member	**<u>Chirag</u>** Front end->notes: add, delete, update, view all Front end-> navigation bar **<u>Koosha</u>** Client meeting Back end->notes: add, delete, update, view all Back end->user: login, signup, session Back end->user: define model Backend->middleware setup for all functions Authentication for user: bhash, jsonwebtoken **<u>Tash</u>** Repo initial set up: react app, app.js, connect to db Front end->contact: add, delete, update, view all Front end->login, sign up Front end-> navigation bar Back end->contact: add, delete, update, view all **<u>Olivia</u>** Back end->meetings: add, delete, update, view all Back end->define model for contact, notes, meetings, order Sprint 1 user stories draft and final Domain class diagram **<u>Anagha</u>** Client meeting and communications Back end->contact: add, delete, update, view all **<u>Team</u>** Troubleshooting and bug fixing
Sprint achievements	User functionalities: (back end, front end, connected via middleware) • Add, delete, update, view all contacts • Add, delete, update, view all notes • Login, signup • Authentication

Backlog items, status	Implement password rules: in review Unit testing for notes controllers: not assigned Unit testing for sign in/sign up controller: not assigned
Backlog items from previous sprint	NA
Backlog items removed from sprint	NA
Link to kanban board	
Link to retrospective	Development Sprint 1

Development Sprint 2

Navigate to	Link
Sprint Planning	Sprint Planning: Development Sprint 2
Sprint Review	Development Sprint 2 - Sprint Review
Sprint Report	Development Sprint 2 - Sprint Report
Retrospective	Retrospective-development sprint 2

Sprint Planning: Development Sprint 2

Trello board for this sprint

By the end of the sprint, users should be able to:

Note: numbers reference user story table entries : User Stories

- 3. Schedule meetings (Olivia)
- 6. Create labels and label contacts (Olivia, Tash)
- 7. View and edit upcoming and past meetings (Olivia, Chirag)
- 8. Create edit and delete notes for meetings (Olivia)
- 9. Send emails (Koosha)
- 10. Create, edit and delete orders (Anagha, Tash)
- 11. Edit user profile

Backlog

- 4. Edit contacts (Tash)

Low Priority (research and consideration when implementing order functionality)

- Data visualization

Tasks:

Labels

1. Controller to add, delete labels (include set colour) (S) S S S S S
2. Controller to get all labels (S) S S S S S
3. Controller to assign a label from drop down to a contact (M) L S S M M
4. Controller for filtering contacts based on labels? (L) XL L L L   L
5. Front end view all labels in a drop down (M) M L M M M
6. Front end select a label for a given contact (M) M S S S   M
7. Front end delete a label for a given contact (S) S S S S   M
8. Front end add a new label to label db (S) S S S S   S
9. Front end delete a label from label db (S) S S S S   S
10. Front end search bar for filtering based on labels (L) L XL M L   L

View and edit upcoming and past meetings (added to Trello)

1. Controller for all other meeting functionality (L) L L L M L

2. Frontend switch to switch between past and upcoming meetings (M) M M M L M
3. Frontend design for each meeting (L) M L L L L
4. Creating Notes from Meetings Page (M) L S M M M
5. Frontend for deleting and editing Meetings (possibly a pop up) (M) M L L M M

Create, edit and delete orders (added to Trello)

1. Controller for adding and deleting orders (M) S M M M M
2. Controller for updating or cancelling orders (L) L L M L M
3. Tracking orders stage (dropdown) (M) M M S M M
4. Filtering orders based on contact (L) L L M L XL
5. Front end design for order page (M) M XL M L M
6. Front end for tracking orders (M) S M M M M
7. Front end for filtering orders (M) M L M M M
8. Potential extension for data visualisation (research) (M) XXXXXL S XL

Edit user profile (added to Trello)

1. Controller for updating user information (M) M M M M M
2. Front end component for editing user profile (M) S M M M M

Send emails (added to Trello)

1. Obtain OAuth 2. 0 credentials (M) M ? M ?
2. Obtain an access token (M) M ? M ?
3. Record email content (M) S M M M
4. Send email using nodemailer and access token (L) M XL M ?

Retrospective-development sprint 2

Date	07 Oct 2021
Team	Dream
Participants	@ Natasha Ireland @ Olivia Ryan @ Anagha Giri @ kowji @ Chirag Charan Singh

Background

The primary goal of this sprint was to produce an email API for user to send emails, label functionality for users to assign labels to contacts, meeting functionality for scheduling meetings and linking them to notes, showing and updating orders and finally updating a user profile.

Retrospective

Start doing	Stop doing	Keep doing
• Resolve all warnings/errors in the code • Maintain a list of package installs maybe so when we pull from main after each update we know what to install • Push clean and polished code • Comment as we go • Write test cases as code is developed • Catching errors in the code (Error-proof code) • Be careful about folder structures before merging to main • A little bit more collaborative	• Multiple people editing on the same branch • Stop pushing node modules to branches • Merging to main too early (merging with bugs)	• Jumping in to help each other with issues • Back and forth communication between front end and back end • Showcasing completed features • Showcasing major (?) errors •

Action items

- Remove unused Node modules
- Improve Code Formatting

- When merging with main
    - Remove all statements triggering warnings before merging to main
    - Remove debugging statements (console logs, etc)

    - Remove commented out statements
    - Complete adequate debugging before merging to main
- Comment along the way and not in one go
- Implement error catching in the front end
- Restructure any branches that haven't been fixed and pull from main / merge to main

- [x] Don't worry be happy

Development Sprint 2 - Sprint Review

Background:

Date: 15/10/2021 at 11:30 am

Attendees: The Dream Team (Anagha, Chirag, Koosha, Natasha and Olivia) and Client (Yiran)

Agenda: Show the client the progress since our previous meeting and to get feedback on the deliverables created during the sprint.

Meeting Minutes:

Refer to link below for in-depth meeting minutes and the recording of meeting.

Client Meetings

Client Response/Feedback:

Commendations:

- Likes password rules and was impressed
- Likes avatar for presentation

Recommendations:

- Would like contacts on order page to drop down as typed
    - Low priority - nice to have rather than requirement for submission
- Would like a pop-up if incorrect password has been entered on login page

Development Sprint 2 - Sprint Report

Dates	6/9/21-4/10/21
Contributions by team member	**Chirag** Front end->meetings Testing->unit testing (notes and contacts), integration testing (notes) **Koosha** Back end and Front end->Email page, profile page Deployment->setting up pipeline **Tash** Front end->labels page, order page **Olivia** Back end->meetings, labels Front end->profile page **Anagha** Client meeting and communications Back end->orders Testing->integration testing (contacts) **Team** Troubleshooting and bug fixing

Sprint achievements	User functionalities: (back end, front end, connected via middleware) • Add, delete, update, view all orders • Add, delete, update, view all meetings • Add, delete, update, view all labels • Send emails • View and update user profile
Backlog items, status	Integration Testing - Contacts: Doing (moved to sprint three)
Backlog items from previous sprint	All completed
Backlog items removed from sprint	NA
Link to kanban board	
Link to retrospective	Retrospective-development sprint 2

Development Sprint 3

Navigate to	Link
Sprint Planning	Sprint Planning: Development Sprint 3
Sprint Review	TBC
Sprint Report	TBC
Retrospective	TBC

Sprint Planning: Development Sprint 3

Addressing user stories: 12, 13, 14, 15, 16, 17, 18

User Stories

Tasks:

1. Testing:

- Unit Testing - CHIRAG
    - Notes and Contacts Edit
    - User.js
    - Email.js
    - Order.js
    - Meeting.js
- Integration Testing - ANAGHA
    - Sign in Page
    - Orders Page
    - Meetings Page
    - Email Page ??
    - Contacts Page
    - Notes Page
- End to End Testing (Playwright) (XXL)

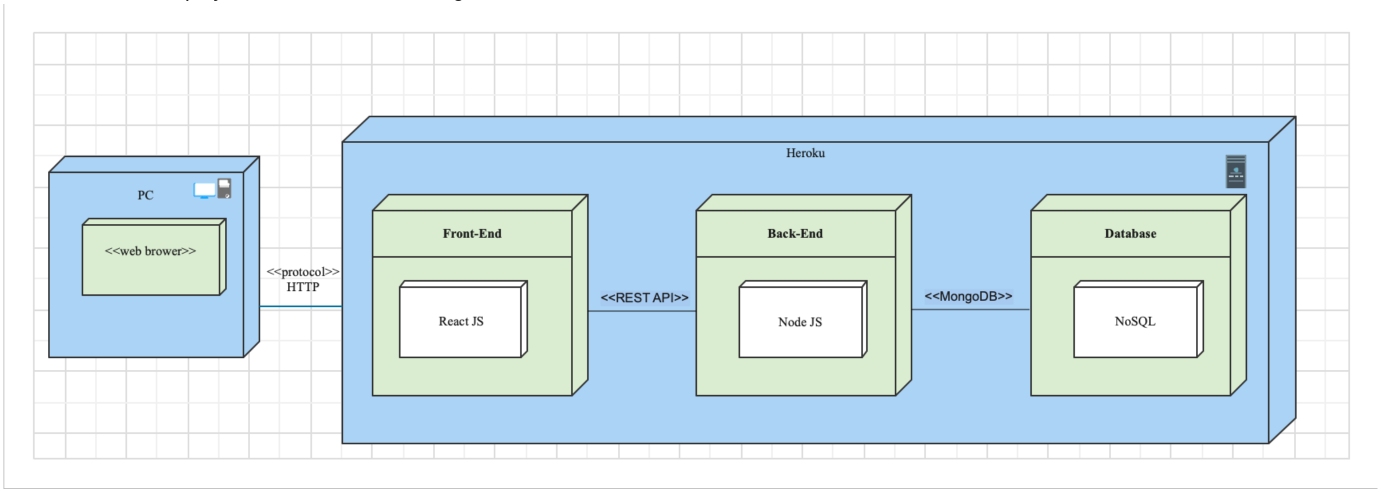
1. Search notes by title frontend (possibly backend) - KOOSHA
2. Filter notes (based on time and alphabetical) frontend (possibly backend) default A-Z (CONFIRM)
3. Search contacts (based on first name) - TASH
4. OPTIONAL (Searching contacts (based on labels)) - LIV
5. Failure page - TASH
6. Profile Page (backend) - changing passwords -KOOSHA
7. Password Requirements Setup - KOOSHA
8. Organise code in stateful components - TASH
9. Presentation - Thu
10. Meetings - add contacts - CHIRAG

Sprint Planing Process

- User stories are split up between 2 development sprints based on priority
- As a group we produce realistic steps to how we can achieve the following user stories for the current development sprint
- As a group we produce an estimation of time and effort for each user story using planing poker and t-shirt sizes
- Each user story with their time and effort estimation is added to our kanban board and assigned to members of our group

Deployment Process

Our team deploys the front-end and back-end of our system to Heroku. This action is done via GitHub Actions. Our GitHub Actions workflow is activated upon pushes to main branch and pull requests and after building the front-end(ReactJs) and back-end(NodeJS) separately and testing both of those builds it automatically deploys to Heroku. When an end user visits our system they connect using http request to our front-end Heroku server. Our front-end and back-end communicate using REST API and our back-end communicates to our database using mongoDB. You can see the deployment workflow in the diagram below.



Quality Assurance

This folder contains all the testing/Deployment information and reports for the project DreamCRM

Navigate to	Link
Testing Plan	Testing Plan
Deployment Process	Deployment Process
Testing Report - Development Sprint 1	Testing Report - Development Sprint 1
Testing Report - Development Sprint 2	Testing Report - Development Sprint 2
Testing Report - Development Sprint 3	Testing Report - Development Sprint 3

Testing Plan

Objective

The primary objective of this testing plan is to prevent defects and critical bugs in the DreamCRM project. Additionally, to produce documentation that describes the approach and the testing standards. To ensure that most of the components are tested, the coverage is set to be at 70%.

The table below contains the details for the types of testing conducted during this project :

Type	Framework	What to test?	When?	Who?
Unit Testing	• Mocha • Chai • Sinon • Faker	Testing individual functions with successful and unsuccessful cases.	Before merge to main	Team Member responsible for writing controllers
Integration Testing	• Mocha • Chai • Supertest	Testing the connection between the backend controllers, models, routes and the server for successful and unsuccessful cases	After connecting backend and frontend components	@ Chirag Charan Singh @ Anagha Giri

End to End Testing	User Simulation	Simulate the behaviour of a normal user and go through every functionality of the website ensuring that they work as expected.	Before merge to main	All Team Members
Quality Assurance Testing	Client	Testing the quality standards of the final product	Upon final product delivery	Client (Yiran)

Testing Reports

Navigate to	Link
Development Sprint 1	Testing Report - Development Sprint 1
Development Sprint 2	Testing Report - Development Sprint 2
Development Sprint 3	Testing Report - Development Sprint 3

Testing Report - Development Sprint 1

Testing Reports

- End to End Testing

Test	Result	Notes
Create a new Account	Success	
Log in an already existing account	Success	
Create a contact	Success	
Edit a contact	Success	
Delete a contact	Success	
Create a new note	Success	
Edit a note	Success	
Delete a note	Success	
Logout of account	Success	

- Unit Testing

Test	Result	Notes
Get all Notes	Success	
Delete Note	Success	
Create Note	Success	
Get all Contacts	Success	
Delete Contacts	Success	
Create Contacts	Success	

- Behaviour-driven development / Acceptance test-driven development
- As a user, I want to be to navigate to contacts and notes easily and quickly.

Accepted Criteria: - A user should be able to access contacts and notes from any page using the navigation bar. This feature was also presented to the client, who also appreciated the simplicity and intuitiveness of this navbar.

Testing Report - Development Sprint 2

Testing Reports

- End to End Testing

Test	Result	Notes
Create a Meeting	Success	
Edit Meeting	Success	
Assign Contact to a Meeting	Success	
Mark Meeting as Completed	Success	
Create a Meeting Note	Success	
View Meeting Note on Notes Page	Success	
Edit Meeting Note	Success	
Delete Meeting	Success	
Create Order	Success	
Update Order	Success	
Delete Order	Success	
Assign Contact to Order	Success	
Update Profile Details	Success	
Update Password	Success	

- Integration Testing

Test	Result	Notes
Log in an already existing account	Success	
Create a new Account	Success	
Get all Notes	Success	
Create Notes	Success	
Update Notes	Success	
Delete Notes	Success	
Get Meeting Note	Success	
Get all Meetings	Success	
Create Meetings	Success	
Update Meetings	Success	
Mark Meeting as Completed	Success	
Delete Meeting	Success	

Testing Report - Development Sprint 3

Testing Reports

- End to End Testing

Test	Result	Notes
Search Contacts	Success	

Filter Contacts	Success	
Search Notes	Success	

- Integration Testing

Test	Result	Notes
Get all contacts	Success	
Create contact	Success	
Update contact	Success	
Delete contact	Success	

- Unit Testing

Test	Result	Notes
Create label	Success	
Delete label	Success	
Get all labels	Success	
Create meeting	Success	
Delete meeting	Success	
Get all meetings	In progress	
Update meeting	In progress	
Create order	Success	
Get all orders	Success	
Delete order	Success	

Style and Conventions

This page outlines the style and conventions to be standardised across the project relating to coding and git use.

Coding Style

Frontend: camel case

Backend: snake case

File Naming Conventions

Backend: filenames contain the feature name and NOT the folder name

eg. For the order controller: order.js

Frontend: filenames contain the feature name followed by “.component”

eg. For the orders page: OrderPage.component.js

Branch Naming Conventions

Feature branch: FEAT-<feature name>

Individual dev branch: <name>-dev

Commenting Conventions

At the Beginning of Each File

Structure

```
/**
 * <name of file>, <Purpose of file>
 * Created for IT Project COMP30022, Semester 2 2021
 * The University of Melbourne
 * <Code rights or taken from> (if applicable)
 * Implemented by DreamTeam: Anagha Giri, Koosha Owji, Chirag Singh,
Olivia Ryan, Natasha Ireland
 */
```

Eg.

```
/**
 * label.js, Contains backend functions relating to labels.
 * Created for IT Project COMP30022, Semester 2 2021
 * The University of Melbourne
 * Implemented by DreamTeam: Anagha Giri, Koosha Owji, Chirag Singh,
Olivia Ryan, Natasha Ireland
 */
```

Before functions

//Input: <Input> Output: <Output>

//Purpose of function

Structure

```
/**
 * Function purpose
 * @param {<desc>} <param>
 * @param {<desc>} <param>
 ...
 * @returns {<desc>}
 */
```

Eg.

```
/**
 * Given a contact id and a label id, insert the label id into a
labelId array stored
 * in the contact.
 * @param {request with "label_id" in the body and "id" (the
contact_id) in the params} req
 * @param {response by which the updated contact json object will be
sent} res
 * @returns {the response}
 */
```

Within code commenting

Structure

```
// <Anything that is unobvious to explain code>
```

Eg.

```
// Check for existing labels with the same title and colour.
```

Meeting notes

This folder contains the notes and recordings from the meetings conducted with the client and supervisor. It also contains any meeting minutes of the meetings among the team members.

Navigate to	Link
Client Meetings	Client Meetings
Supervisor Meetings	Supervisor Check Ins
Team Standups	Standups

Supervisor Check Ins

Abhisha Check In's

Date	Attendees	Agenda	To Do
2/8/21	• Abhisha • Anagha • Chirag • Koosha • Natasha • Olivia	> first catchup > progress report	> assign roles > research CRM > formulate questions > contact client for first meeting > set up repositories

9/8/21	• Abhisha • Anagha • Chirag • Koosha • Natasha • Olivia	> check that Abhisha has access to Trello, confluence and git repo > progress report - met with client, started the domain model, started using Adobe to create some sample UI designs	> User stories and personas > DB schema > Software design diagram > Showcase the front end prototype to the client
16/8/21	• Abhisha • Anagha • Chirag • Koosha • Natasha • Olivia	> progress report	> sprint planning > set up git repo > motivational model
23/8/21	• Abhisha • Anagha • Chirag • Koosha • Natasha • Olivia	> progress report	> product roadmap
30/8/21	• Abhisha • Anagha • Chirag • Koosha • Natasha • Olivia	> progress report	> home page on confluence > product roadmap > pipeline
6/9/2021	• Abhisha • Anagha • Chirag • Koosha • Natasha • Olivia	> progress report	> review meetings pages, design model, motivational model - feedback from Checklist 1
13/9/2021	• Abhisha • Anagha • Chirag • Koosha • Natasha • Olivia	> progress report	> feedback for peers > sprint documentation - sprint reports, reviews and planning
27/9/2021	• Abhisha • Anagha • Chirag • Koosha • Natasha • Olivia	> progress report	> deployment pipeline > review feedback from team members
4/10/2021	• Abhisha • Anagha • Chirag • Natasha • Olivia	> progress report	> sprint retrospective > sprint planning
11/10/2021	• Abhisha • Anagha • Chirag • Koosha • Natasha • Olivia	> progress report	> amend testing plan as per feedback from week 9 checklist > presentation slides

Standups

Date	Attendee	Agenda	To-Do
09/08/21	• All team members	• formalise a list of questions for the client • discuss database software and programming languages • go over domain and design diagrams ◦ notes function - only for meetings or are we including a to-do functionality with random notes?	
16/08/21	• All team members	• formalise branching protocol • formalise feature naming protocol • go through sprint plan	Email API research. REACT PD work for Chirag and Tash.

23/8/21	• All team members	• Sprint planning • motivational model discussion	Start picking up tickets and working on them. Identify blockers for future standups.
30/8/21	• All team members	• product roadmap	Create home page. Create product roadmap.
7/9/21	• All team members	• first dev sprint due • go over notes, contact, login, sign up and nav function delivery	• finish contact update functionality (Tash) • sprint 3 planning • sprint 1 and 2 retrospectives
13/9/21	• All team members	• go over feedback from inception checklist results • discuss admin tasks to be done going forward • outline what everyone is working on going forward	• sprint report for inception and dev 1 (Tash) • sprint planning process write up (Koosha) • client sprint review write up (Anagha) • set up git actions flow • set up some simple tests to be added to flow (Chirag) • deploy to Heroku (to be assigned)

23/9/21	• All team members	• discuss progress since last stand-up
		• go through any blocks in individual tasks
		    • orders page: revisited necessary fields
		    • meetings page: note section to be reworked
			• continue working on individual tasks for sprint as seen on Trello

27/9/21	• All team members	• progress report and see what is left for sprint	• continue working on individual tasks for sprint as seen on Trello
30/9/21	• All team members	• go over requirements for the week 9 sprint checklist to make sure we have requirements	• amend confluence pages to provide more information on testing and deployment
4/10/21	• All team members	• sprint planning for last dev sprint • see what tasks are on backlog	• assign ourselves to tasks on trello
7/10/21	• All team members	• sprint retrospective	• continue working on individual tasks for sprint as seen on Trello • set up meeting with client for sprint review
11/10/21	• All team members	• progress report • fix bugs with deployment	• work on individual sections for presentation
14/10/21	• All team members	• presentation slides and allocation of time for each part of presentation	• complete presentation and practice
16/10/21	• All team members	• practice run through of presentation	• continue to practice presentation and amend presentation to ensure that time limit is maintained

18/10/21	• all team members	• trouble shoot contact filter and search • identify final jobs to work on for report and checklist hand in	• update README (look at Abhisha's for example) • boasting page • everyone: report, individual contribution statement, professional communication report: lecture review, ethical analysis • get handover date from Yiren • find out if Yiren wants a user manual • write unit tests (everyone) • fix up testing plan for checklist • error page testing (tash) • contact search and filter (Chirag, Tash) • overall checklist Nov 12th • product checklist: • integration testing finalise (Chirag, Anagha) • unit test: tash, liv-labels, anagha-orders, liv-meetings, Koosha-login, sign up and email, profile • tash-add a label not refresh • delete contact: remove name from order • when you add an order: assign contact id, get contact and save first name and last name

21/10/21	• All team members	• go through changes to meetings and contact pages
		• discuss details for checklist submission and handover
			• Checklist (Week 12)
			  ◦ Update Trello
			  ◦ Update Testing Page
			  ◦ Update Deployment Page and deployment link on home page
			• Export presentation slides for submission
			• Handover
			  ◦ Confirm handover date - email Yiran
			  ◦ Complete user manual
			  ◦ update github readme
			  ◦ licensing agreements
			  ◦ avatars
			• Final checklist
			  ◦ boasting page

Client Meetings

Date	Attendees	Agenda	Notes	Recordings
06/08/21	• Yiran (Client) • Anagha • Chirag • Koosha • Natasha • Olivia	• product overview • functional requirements • non-functional requirements	• wants to be able to create labels which will allow for filtration based on characteristics of the clients • wishes to be able to directly email and set up meeting with clients • prefers web application - but is our preference	
13/08/21	• Yiran (Client) • Anagha • Koosha	• Clarification of functional requirements	• wants to manually enter order details and attach clients to it (add details such as a deadline to the order) • order tracking will include a few stages of the order where the user can change between (these stages can be predetermined such as "in design, development, delivered") • there will only be one product per order (assume the product is a software) • orders are only outgoing (no suppliers) • visualisation is very low priority and doesn't need to be customisable (graph of last month orders, or last year orders) • No need for labels in meetings	zoom_0 .mp4

10/09/21	- Yiran (Client) - Anagha - Chirag - Koosha - Natasha - Olivia	Show progress from sprint 1 - login, signup, contacts and notes	Meeting Minutes: - Meeting started at 11:30 am - Explained user login, sign-up, adding contacts, deleting contacts, updating contacts, adding notes, deleting notes and updating notes - Showed preliminary layout for meetings page - Confirmed that next update to client will be provided once next sprint is over Client Comments: - Likes the confirmation pop-up when contact is being deleted - Likes the layout for meetings - Would like to make first name and email mandatory for contacts - Likes the colour and format of the website - even if the colour is a bit different to original UI design shown - Would like a filter search bar - Notes: High Priority - search by title - Contact: Low Priority - search by name - Would like to be able to order notes by either time or title (alphabetical) Low Priority	Sprint1 Client Meeting.mp4
15/10/21	- Yiran (Client) - Anagha - Chirag - Koosha - Natasha - Olivia	- Show progress from dev sprint 2 - orders, meetings, email and profile pages - Get feedback on deliverables	Meeting Minutes: - Meeting started at 11:30 am - Showed a demonstration of functionalities - orders, meetings, email and profile page - Showed updates to password rules - showed UI designs and domain model to see how the product matches original team intention - Client is happy to do User Acceptance Test if required - Confirmed to have a final meeting with her prior to handover Client Comments: - likes password rules and was impressed - likes avatar for presentation - would like contacts on order page to drop down as typed - low priority - nice to have rather than requirement for submission - would like a pop-up if incorrect password has been entered on login page	zoom_0.mp4

Questions

What information about the CRM user needs to be captured?

Does user information need to be update-able?

Preliminary Research

Questions for the client:

Functional Requirements

- What features is the CRM required to have?
- Given this set of features {maybe list features listed below}, which are top priority and which would you consider “out-of-scope”?
- Will the CRM be required to facilitate connections with businesses/services as well as individuals?
- Do you currently use a CRM? If so are there any pain points we should focus on? Have you used any previously - what did you like /dislike about those?

Non-functional Requirements

- Would you prefer a mobile application or a web application?
- What device will be primarily used to interact with CRM functionality?
- Does the CRM need to support multiple languages?
- What sort of traffic will the site be expected to handle? **Might need to rephrase this**
- Speed requirements?
- What are some business goals that will be aided by the CRM? (E.g. data tracking, sales etc.) **This may relate to business CRMs rather than personal ones**

CRM Summary

CRM: Customer/personal Relationship Manager

We'll be creating a personal CRM rather than a business one.

Personal CRM: a tool that is designed to help you manage personal relationships (i.e. calender, reminder and notebook all in one)

More info: https://www.nimble.com/blog/personal-crm/

Some popular CRMs that we can use for reference: Circles, Monica, Airtable, (Source; https://www.zoowho.com/personal-crm/ , https://crm.org/crmland/personal-crm)

Common/Expected features

Things that a CRM needs to do:

- Unify contacts from various different places into one place
- Automatically link contacts with calendar and email
- Auto-enriches contacts with social and other data to save time on data entry
- Replace spreadsheets, post-it notes, and other ineffective ways of contact organization

Source: https://www.nimble.com/blog/personal-crm/

Features per connection:

- Dates (eg. birthday)
- Goals
- Factoids
- Personal details (eg. name and address)
- Communication log (emails + texts)
- Reminders and prompts
- Connection types (eg. friends, family, colleagues)
- Relationships (eg. family members, pets, etc)
- Gifts
- Debts
- Notes

Event-tracking: meetings, catch-ups, etc

Requirements

- Security: personal information and connections info must be kept private. Top Priority
- User friendly: non technical POV
- Customisable: flexible design that allows users to control/alter presentation. Low priority

Resources

Checklist of steps developer can take in early CRM dev stages

https://www.computer.org/publications/tech-news/build-your-career/what-is-a-crm-and-how-to-become-a-crm-developer

Archives

Here are some tips to get you started. You can edit this page to see how it works!

On this page:

1. Create a page

- Click "Create" and select "Blank Page" to create your first page.
- New pages are created as children of the page you are currently viewing.

2. Add to your page

- Click "Edit" to enter the Confluence editor and use the page layouts feature to structure your content using sections and columns.
- Use headings to format your text and drag and drop images into your page to provide visual interest.
- Click "Insert" and select "Other Macros" to add macros for navigation, special formatting and other media.

3. Organise your pages

Here are some tips for organising your content.

- **Change the page order**

The sidebar on the left displays your pages in a hierarchy. If you have Space Administrator permissions you can click "Space Tools" > "Reorder Pages" to move pages around.

- **Add labels**

  Labels help keep pages organised and make it easier for you to find the information you need. Click "Labels" at the bottom of a page to add or edit. The "Related pages" section on this page uses labels too!

- **Make templates**

  Standardise and speed up the page creation process with templates. You can create and format a template with page layouts, standard headings and instructional text for hints and guidelines. Check out our sample page on "Making a template"

Related pages

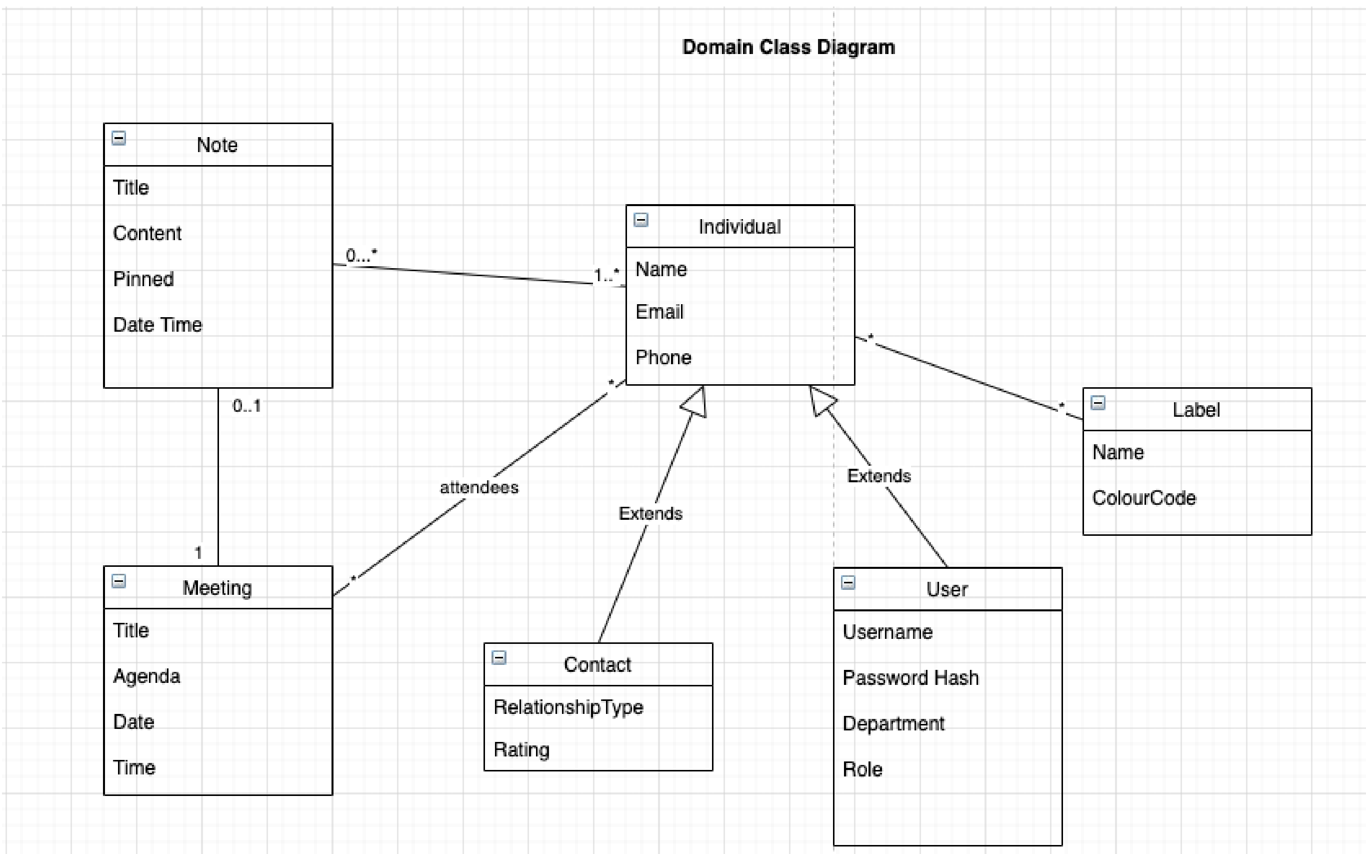
- Archives

Back End

Design Models

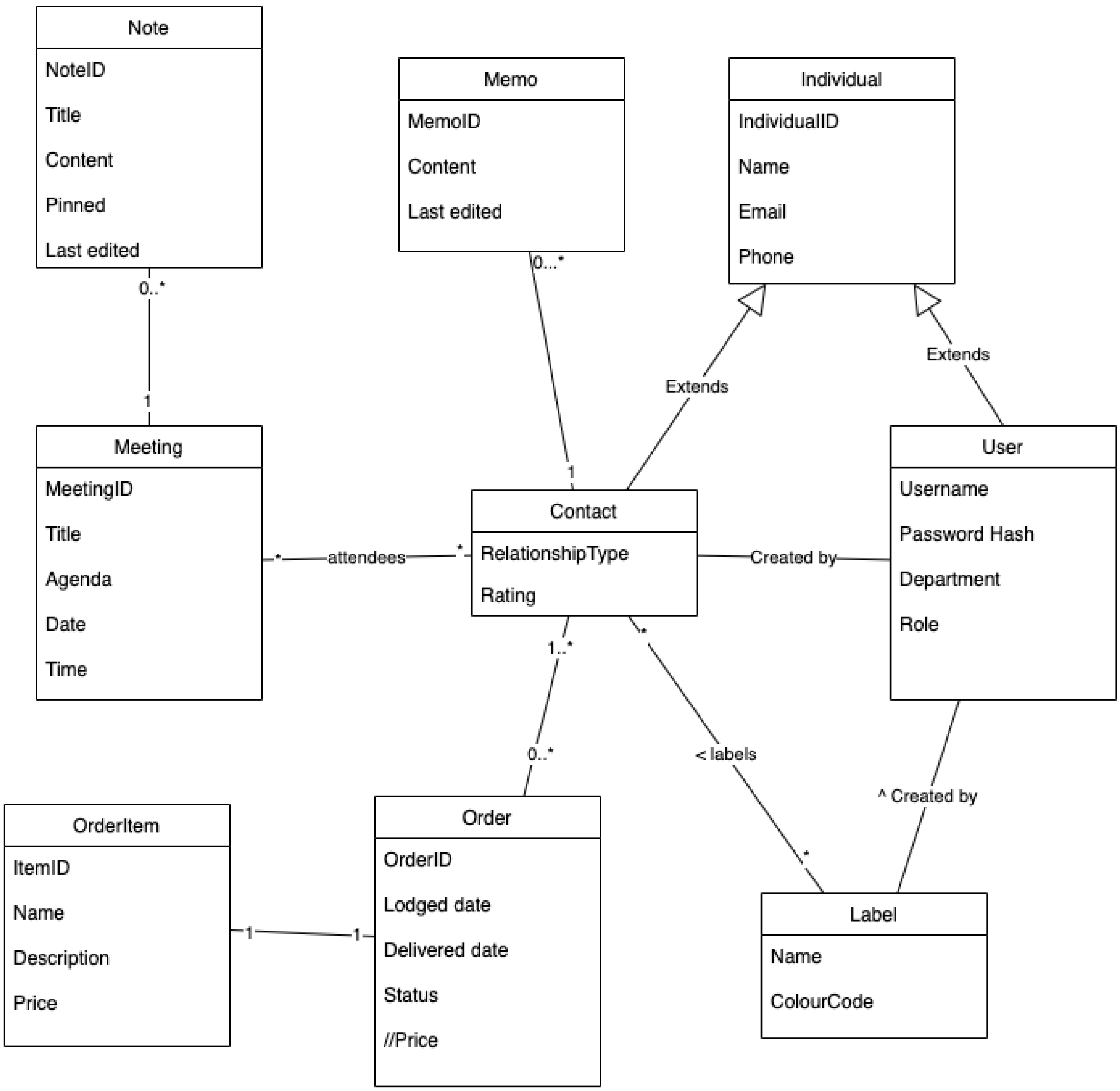
Domain Model - #Iteration 1

View the “Disparities between domain model and front end prototype” section of the front end design choices page and leave feedback through comments.

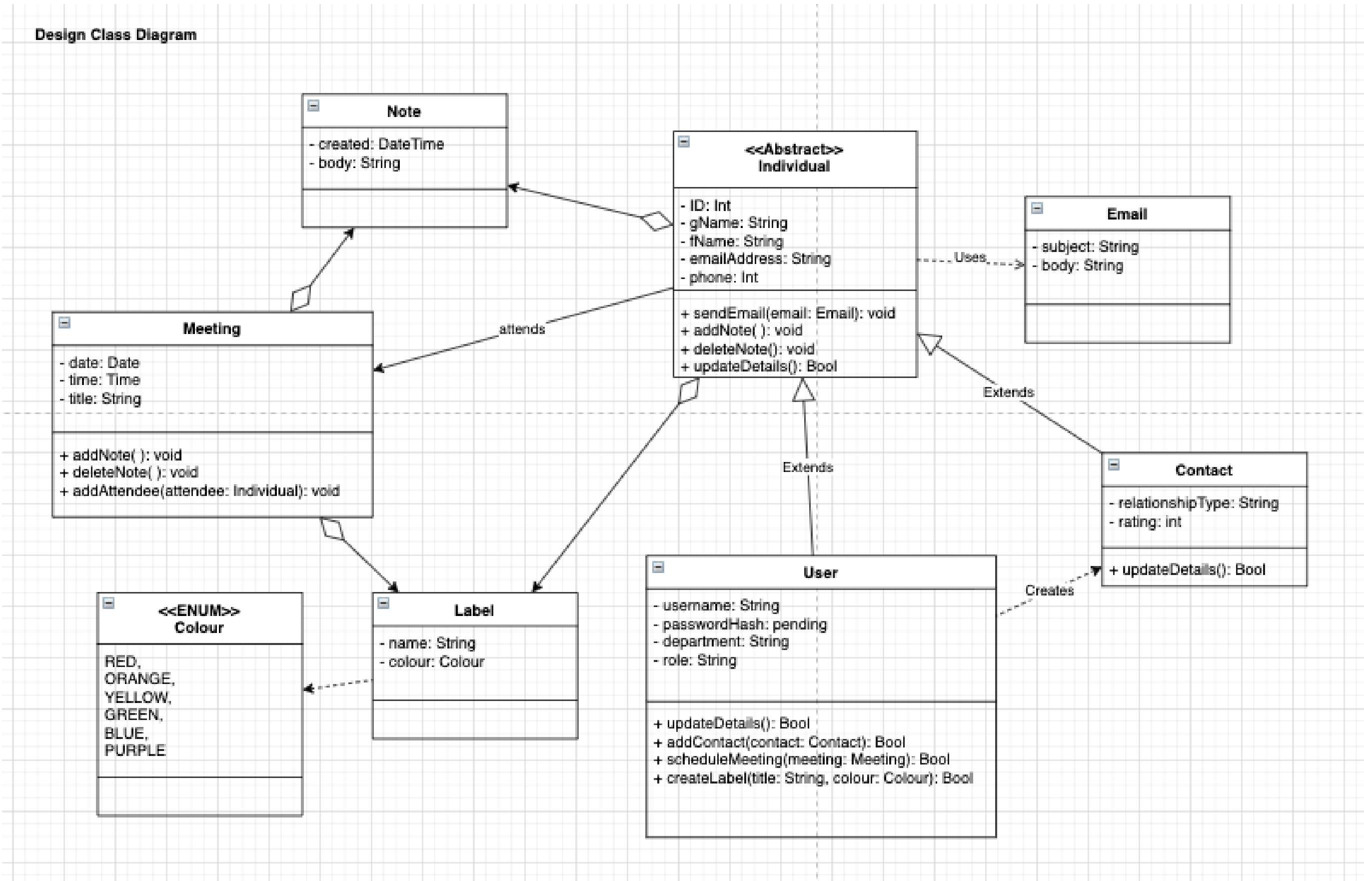


Domain Model #Iteration 2

Domain Class Diagram

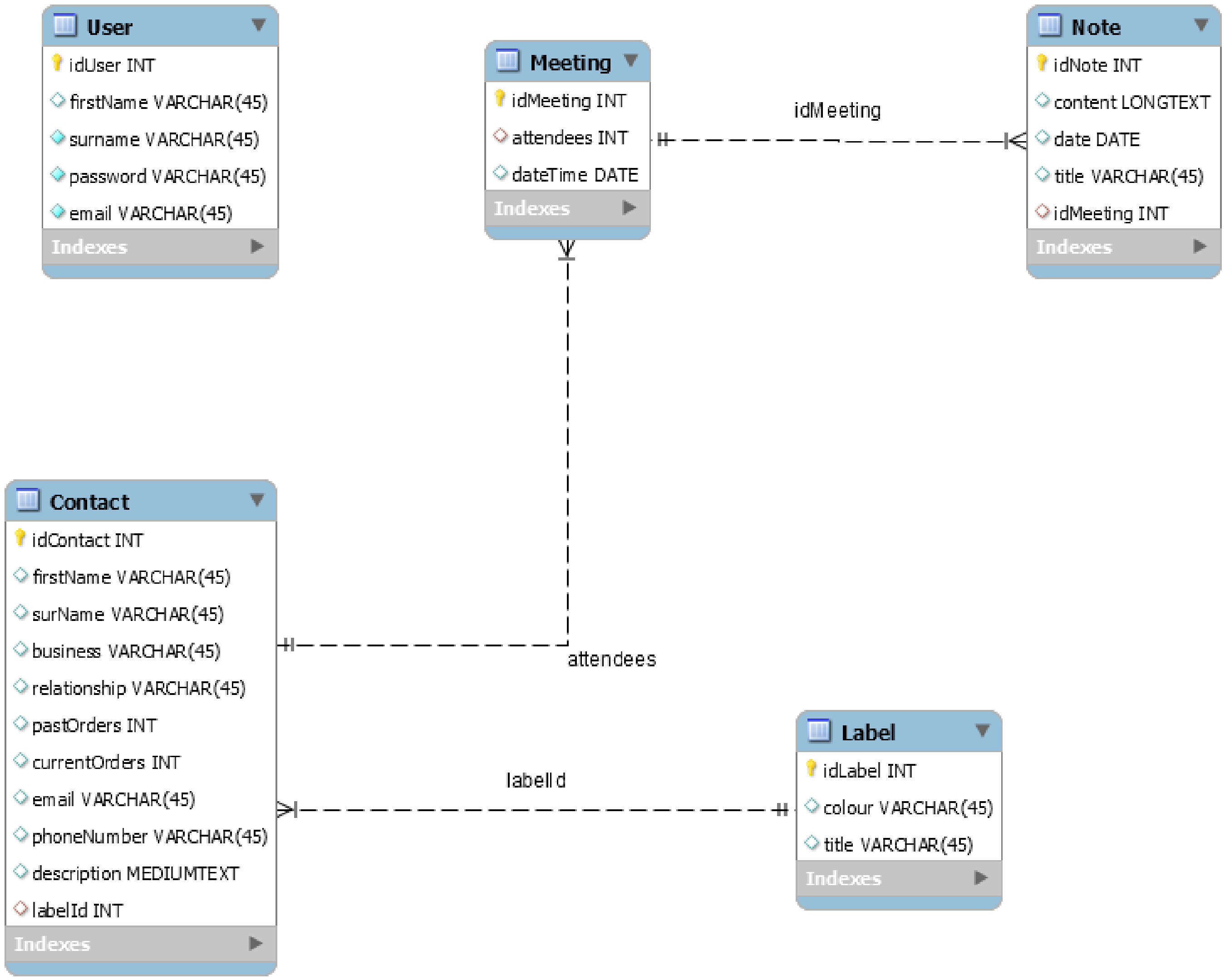


Design Class Diagram #Iteration 1



Database Models/Schemas

Draft in sql just to give a rough overview. We will be using mongoDB so noSQL but can see types and attributes as shown.

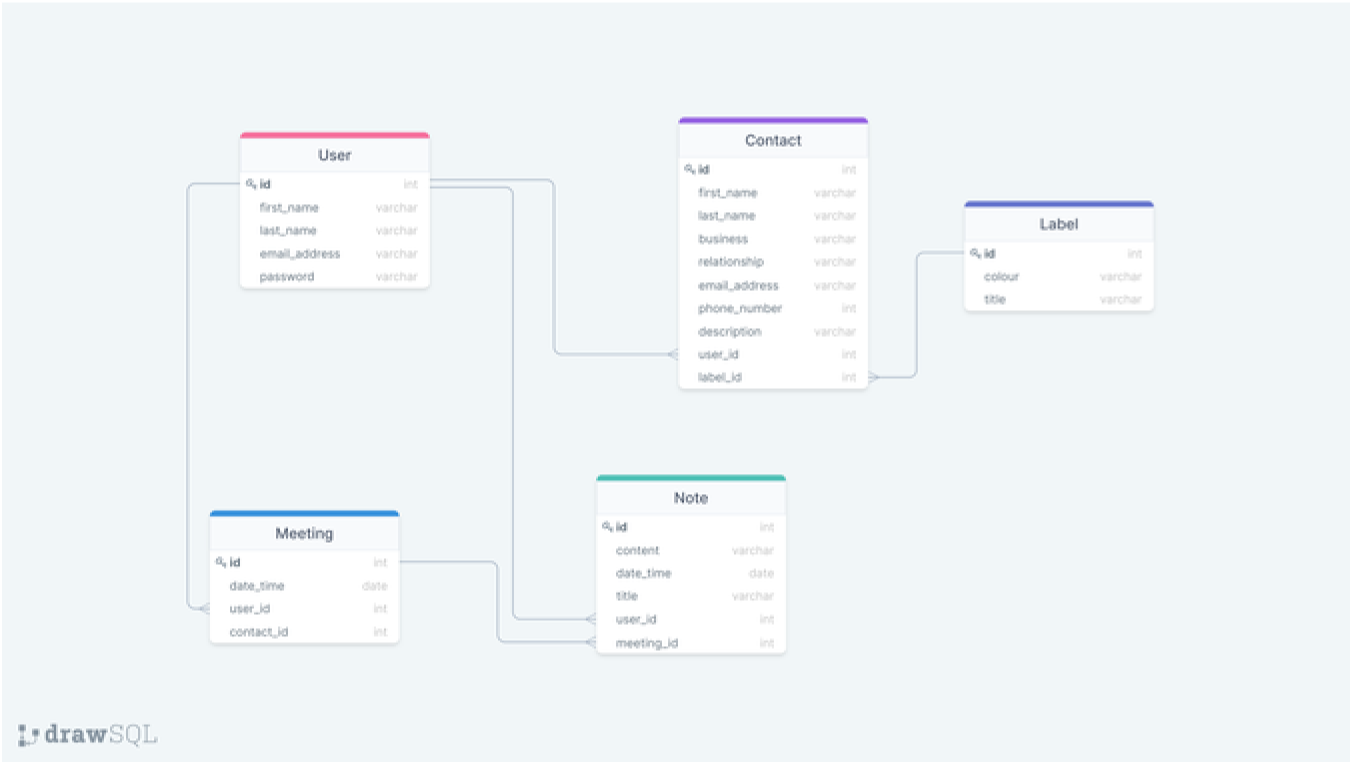


Still iterating over what to do with notes.

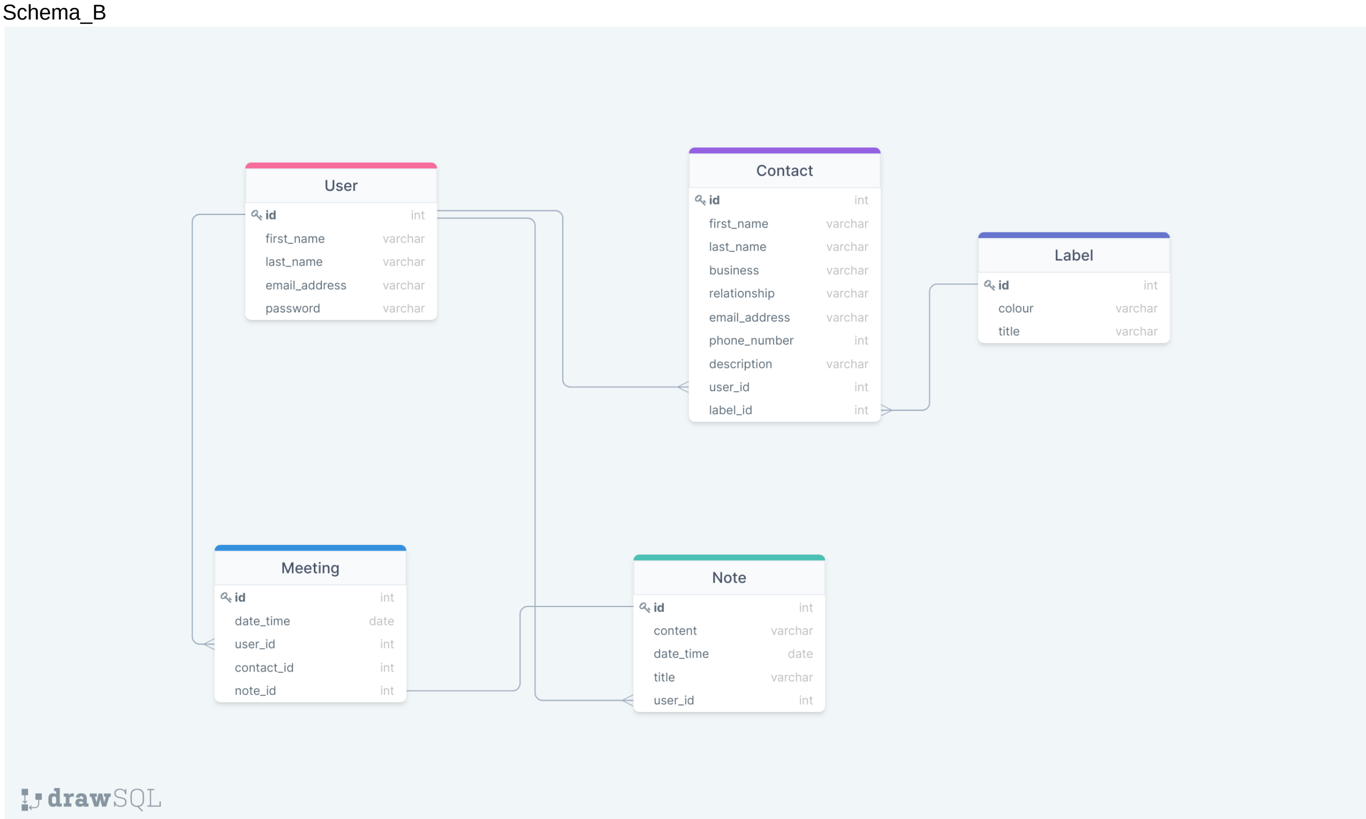
iteration #2

added 3 relationship from User to Contact, Meeting, Note

Schema_A



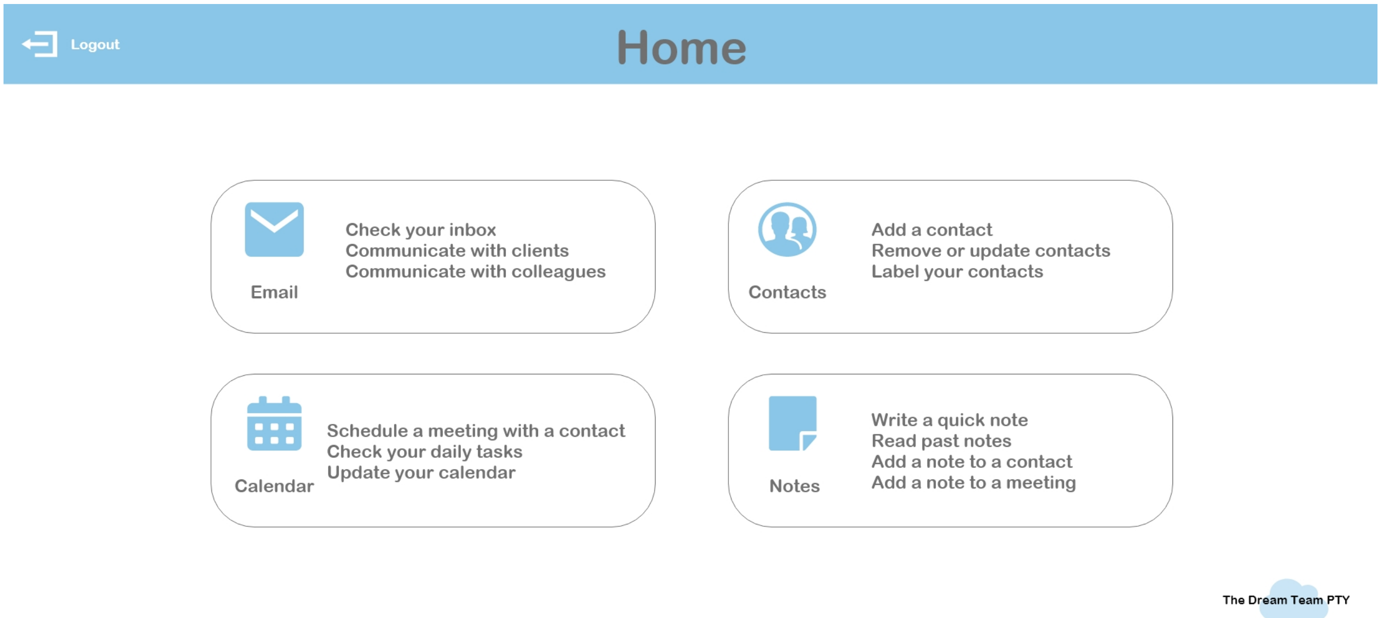
Schema_B

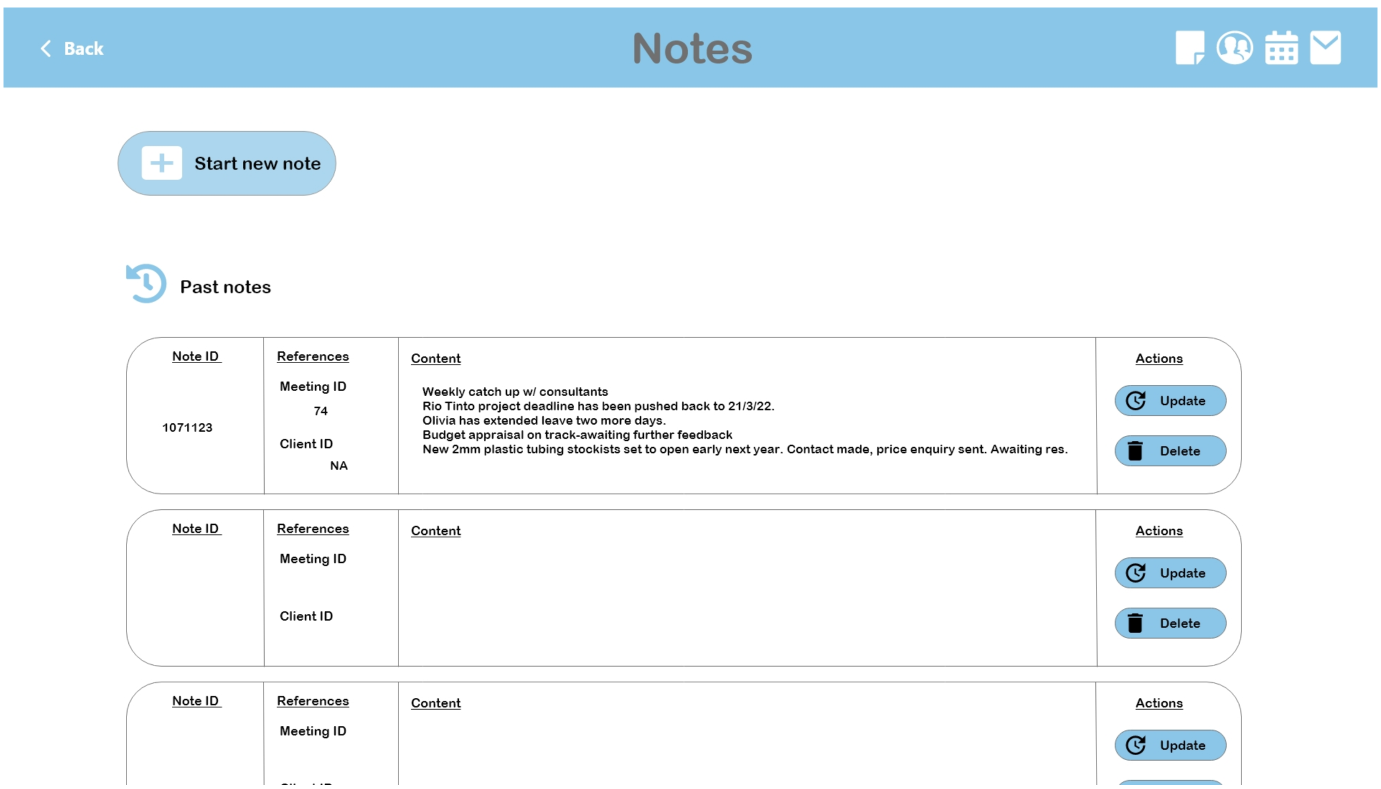


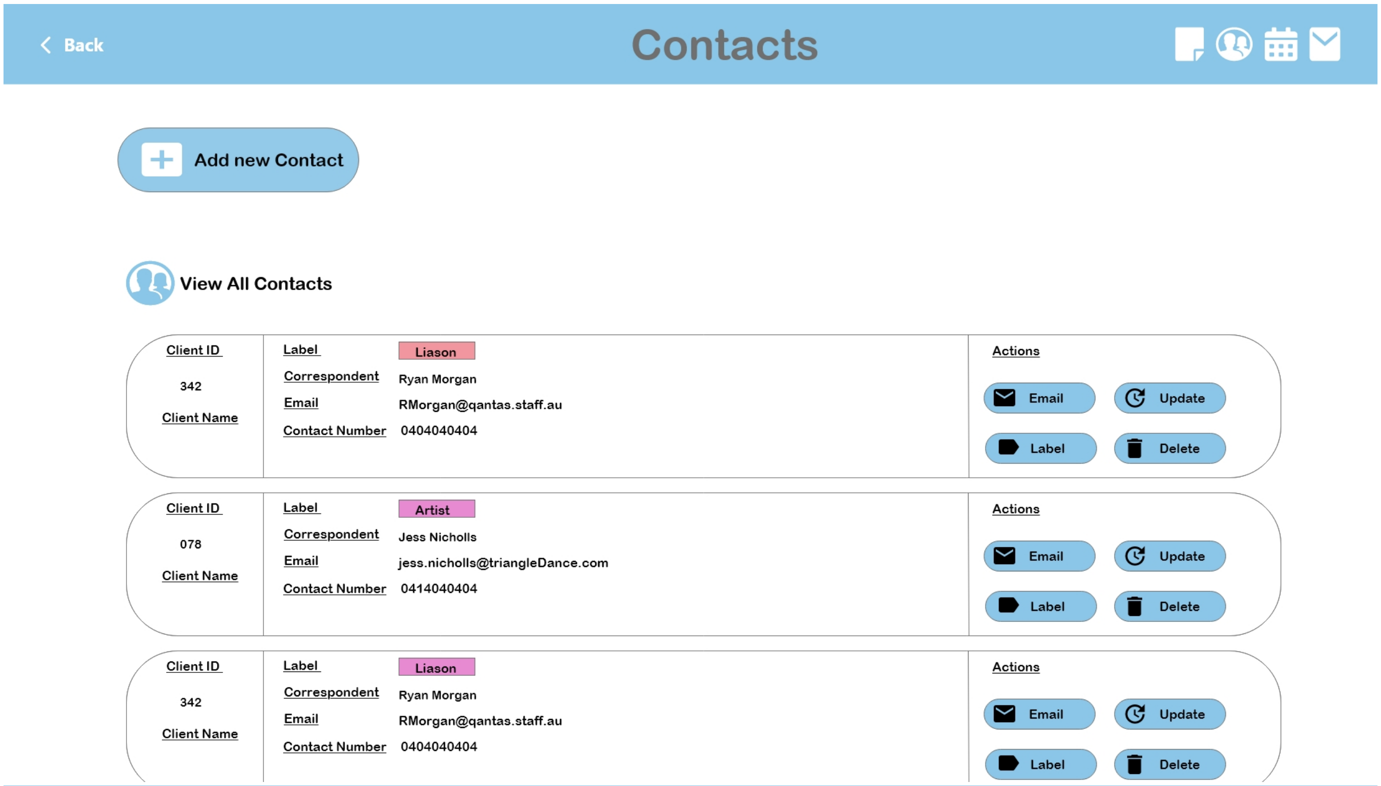
Front End

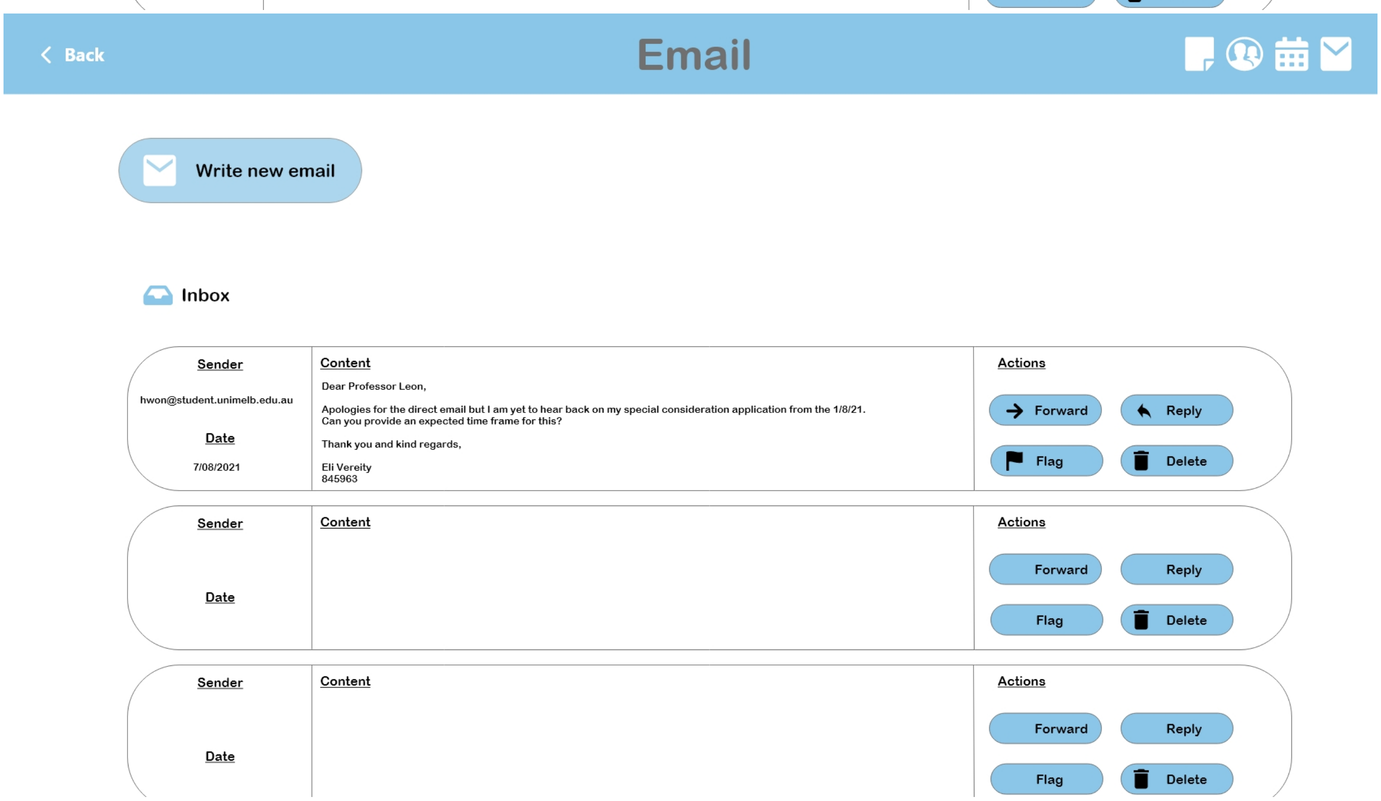
All things beautiful.

UI Design Models


Logout
Home
Check your inbox
Communicate with clients
Communicate with colleagues
Email
Add a contact
Remove or update contacts
Label your contacts
Contacts
Schedule a meeting with a contact
Check your daily tasks
Update your calendar
Calendar
Write a quick note
Read past notes
Add a note to a contact
Add a note to a meeting
Notes
The Dream Team PTY


Back
Notes
Start new note
Past notes
Note ID
1071123
References
Meeting ID
74
Client ID
NA
Content
Weekly catch up w/ consultants
Rio Tinto project deadline has been pushed back to 21/3/22.
Olivia has extended leave two more days.
Budget appraisal on track-awaiting further feedback
New 2mm plastic tubing stockists set to open early next year. Contact made, price enquiry sent. Awaiting res.
Actions
Update
Delete
Note ID
References
Meeting ID
Client ID
Content
Actions
Update
Delete
Note ID
References
Meeting ID
Content
Actions
Update


Back
Contacts
Add new Contact
View All Contacts
Client ID
342
Client Name
Label
Liason
Correspondent
Ryan Morgan
Email
RMorgan@qantas.staff.au
Contact Number
0404040404
Actions
Email
Update
Label
Delete
Client ID
078
Client Name
Label
Artist
Correspondent
Jess Nicholls
Email
jess.nicholls@triangleDance.com
Contact Number
0414040404
Actions
Email
Update
Label
Delete
Client ID
342
Client Name
Label
Liason
Correspondent
Ryan Morgan
Email
RMorgan@qantas.staff.au
Contact Number
0404040404
Actions
Email
Update
Label
Delete


Back
Email
Write new email
Inbox
Sender
hwon@student.unimelb.edu.au
Date
7/08/2021
Content
Dear Professor Leon,
Apologies for the direct email but I am yet to hear back on my special consideration application from the 1/8/21.
Can you provide an expected time frame for this?
Thank you and kind regards,
Eli Vereity
845963
Actions
Forward
Reply
Flag
Delete
Sender
Date
Content
Actions
Forward
Reply
Flag
Delete
Sender
Date
Content
Actions
Forward
Reply
Flag
Delete

‹ Back

Calendar

Schedule an event

Today's Calendar

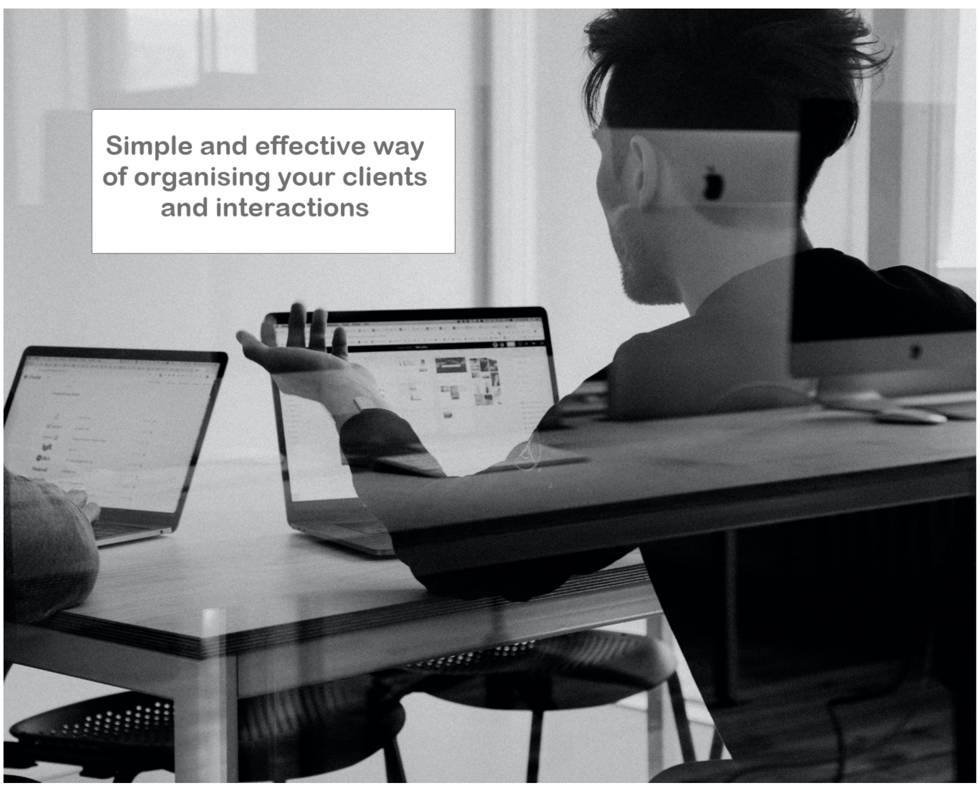
9.00AM meeting with Judy from Production

9.45AM Workplace meditation

1.30PM Susan birthday cake

This Month

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday



Let us (net)work for you

Email

Password

LOGIN

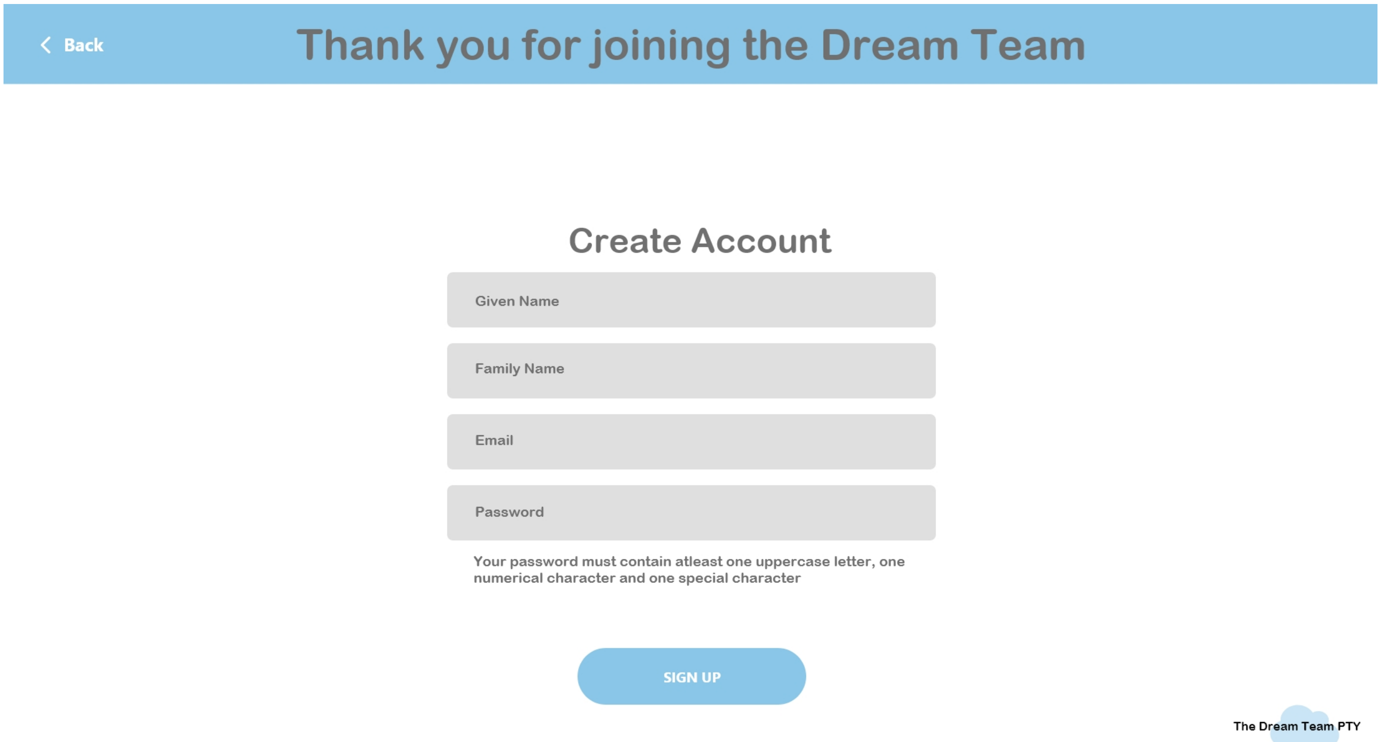
Don't have an account?

SIGN UP

Customer Relationship Management brought to you by

The Dream Team PTY

Intellectual Property of the University of Melbourne



Pages

Home page, client page containing info

Email page where you can

login/create account password. Use password as client details are stored in CRM.

User Flow

Can always access home page using home tag

Appearance

Colours

Fonts

Disparities between domain model and front end prototype:

Elements that require a change based on the resolution are written in red.

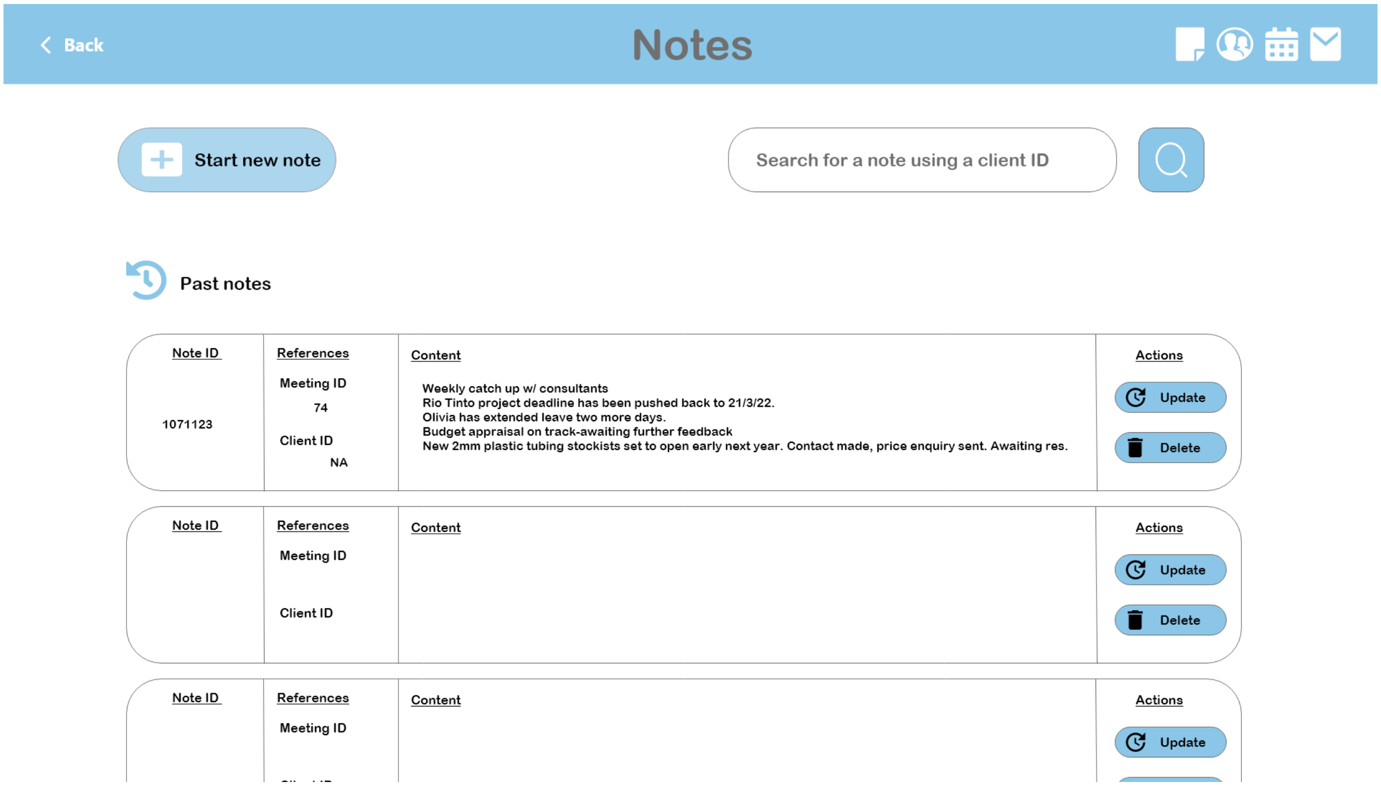
	Domain model	Prototype	Resolution
Notes	1. Pinned attribute 2. Note can reference multiple individuals 3. Notes have optional titles	1. No display of pinned notes 2. Note displays a single individual 3. No title display	1. Unresolved 2. Notes will reference one and only one meeting 3. Unresolved
Contacts	1. No correspondent attribute 2. Multiple labels associated with a single contact	1. Correspondent field 2. Field for a single label	1. Unresolved 2. Multiple labels can be displayed for each contact
Meetings	1. Meetings	1. No meetings display	1. Unresolved
Users	1. Users have a more detailed profile than contacts (ie. role and department attributes included)	1. No display for users / display for contacts doesn't account for additional user info	1. No interaction between users so only account style information need be displayed
Email	1. No email API integration modelled	1. Email functionality displayed	1. Unresolved
Calendar	1. No calendar API integration modelled	1. Calendar functionality displayed	1. Display meetings in a list format for now

Potential API

free standard JS calendar FullCalendar - JavaScript Event Calendar

free email service Free Email API Service for Developers (mailjet.com)

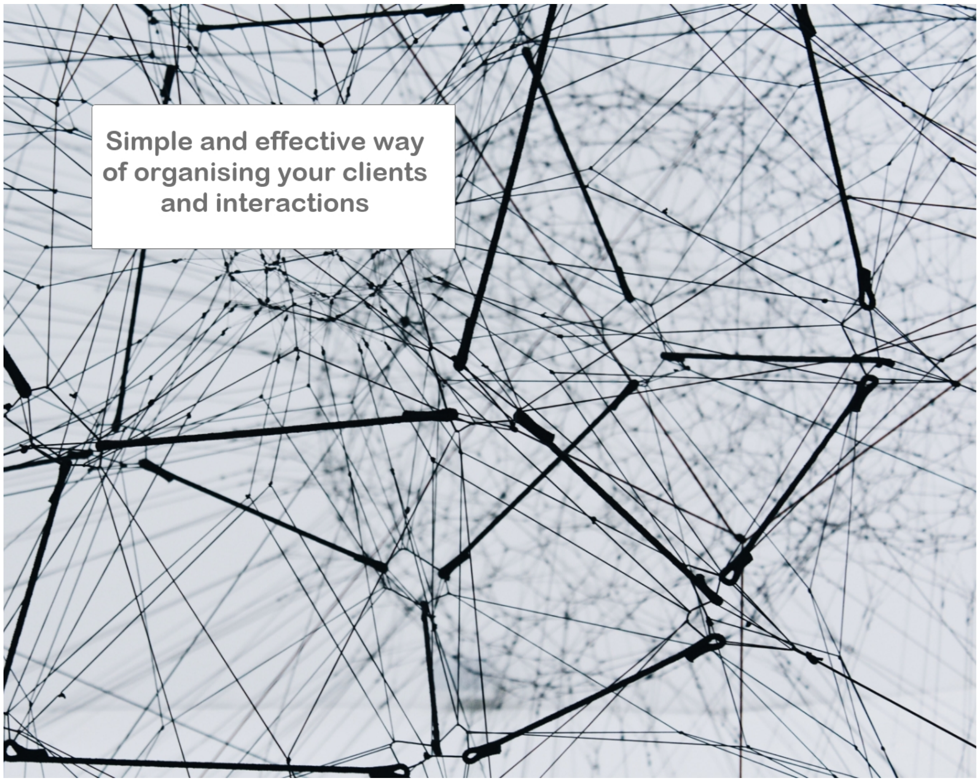
Iteration #2



Use case: user can search in notes to find all related to a client ID: these may have been added from the notes section or meetings section.

Iteration #3

1. I would change the background picture on the login page to show more gender equality
2. In iphone12 - 3.jpg, I also want a field for memo/remark (whatever you name it) to record some important information like background and hobbies.



Let us (net)work for you

Email

Password

LOGIN

Don't have an account?

SIGN UP

Customer Relationship Management brought to you by

The Dream Team PTY

Intellectual Property of the University of Melbourne

Back

Thank you for joining the Dream Team

Create An Account

Given Name

Family Name

Email

Username

Password

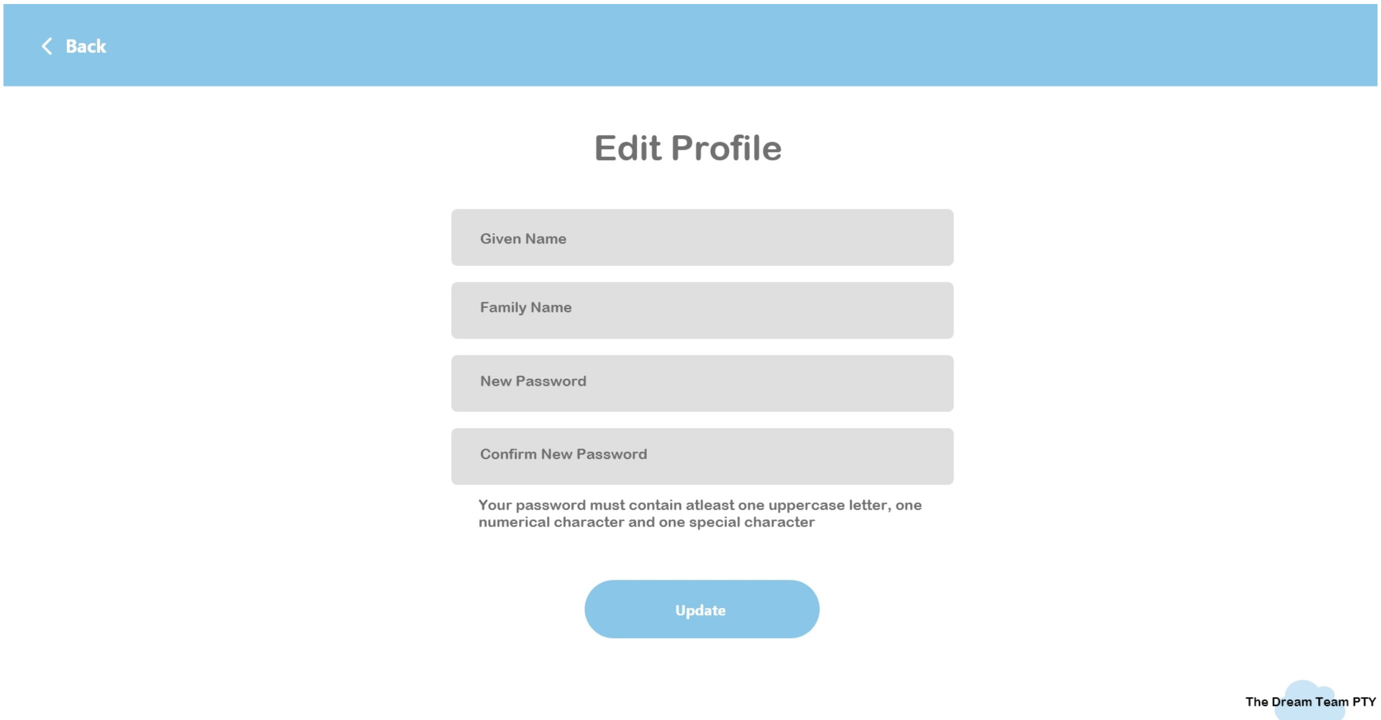
Phone

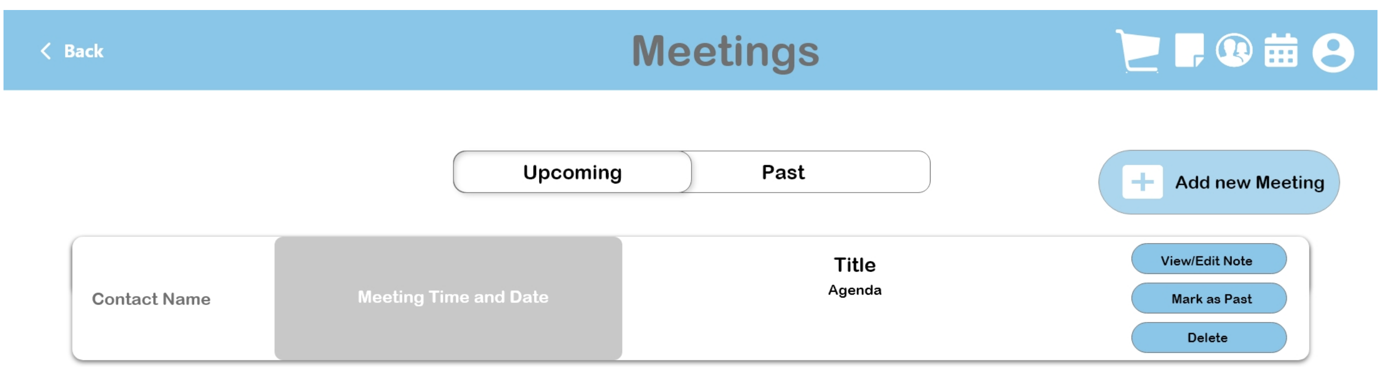
Department/Role etc??????????/?/?????

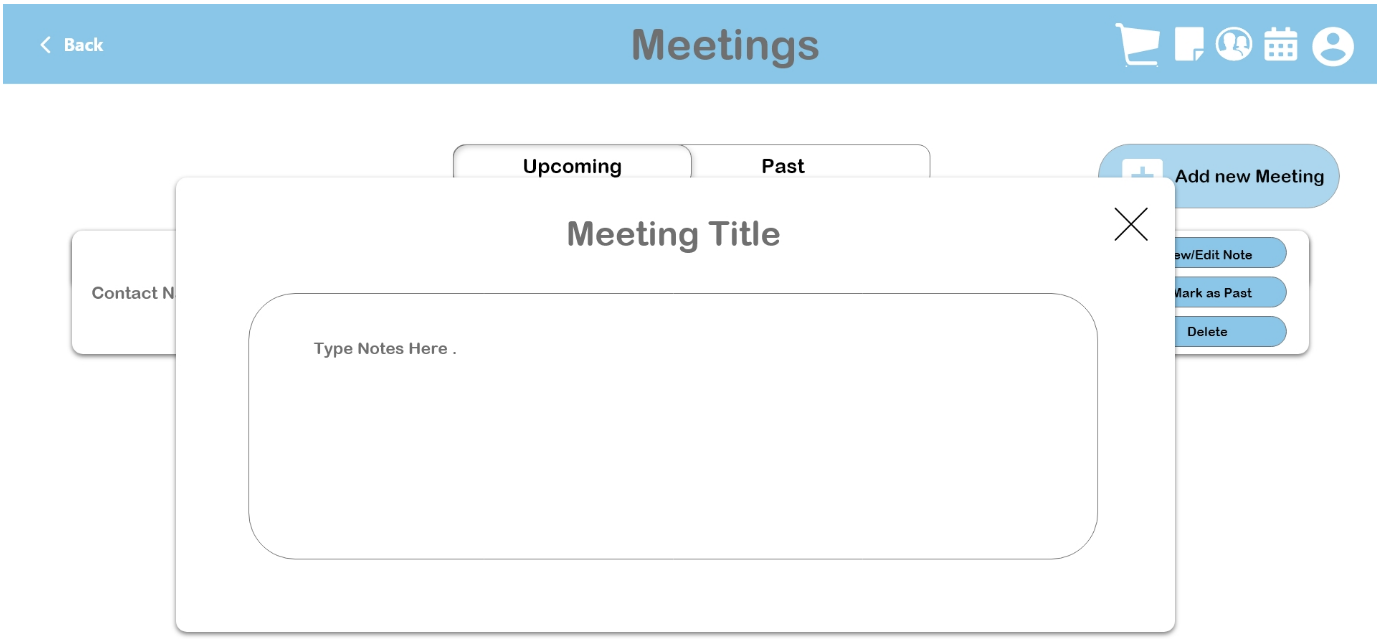
The Dream Team PTY

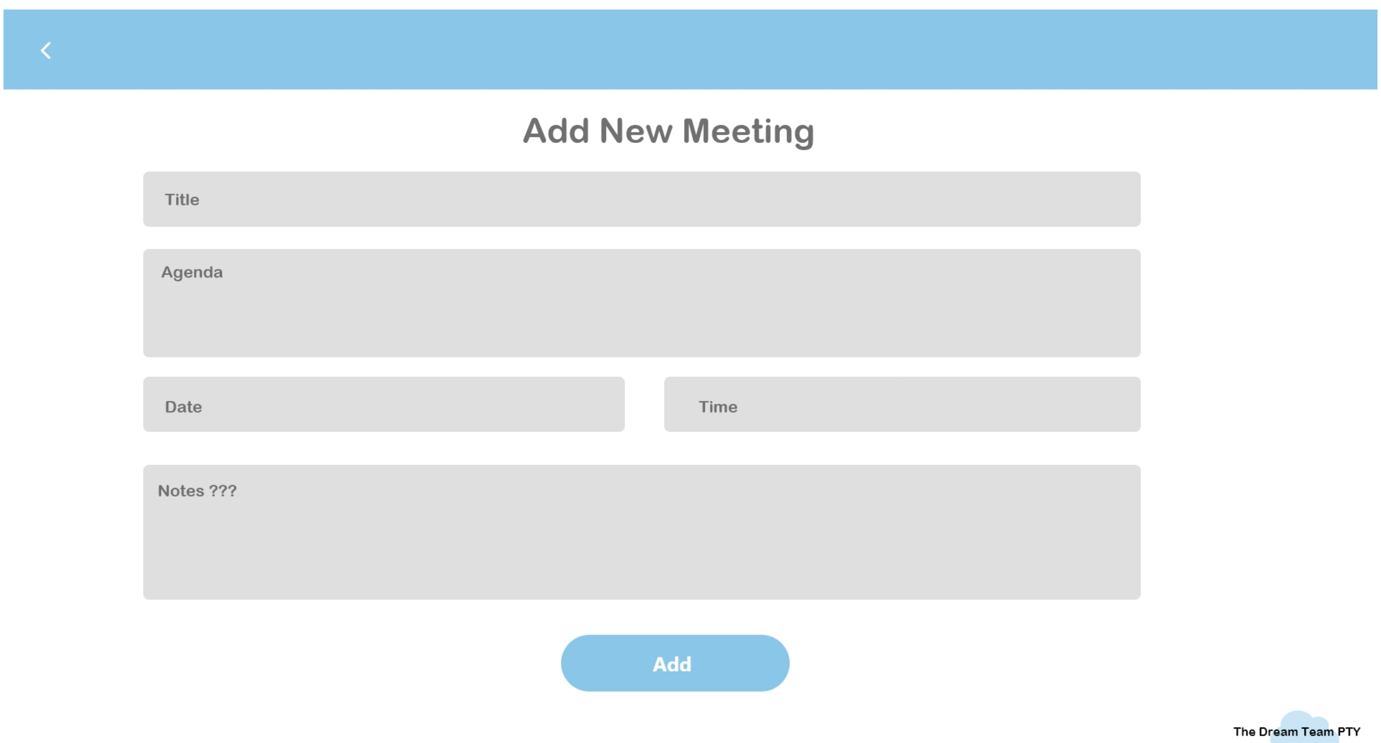
Your password must contain atleast one uppercase letter, one numerical character and one special character

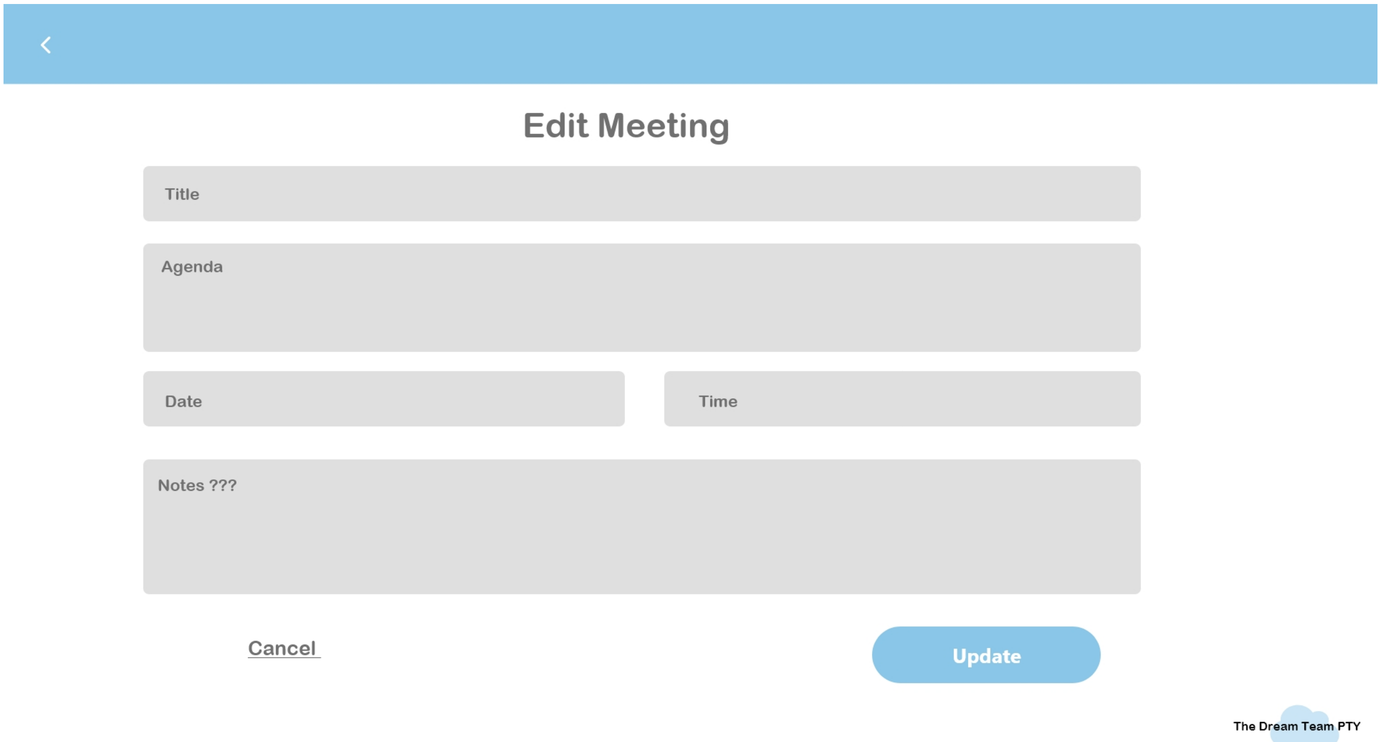
SIGN UP

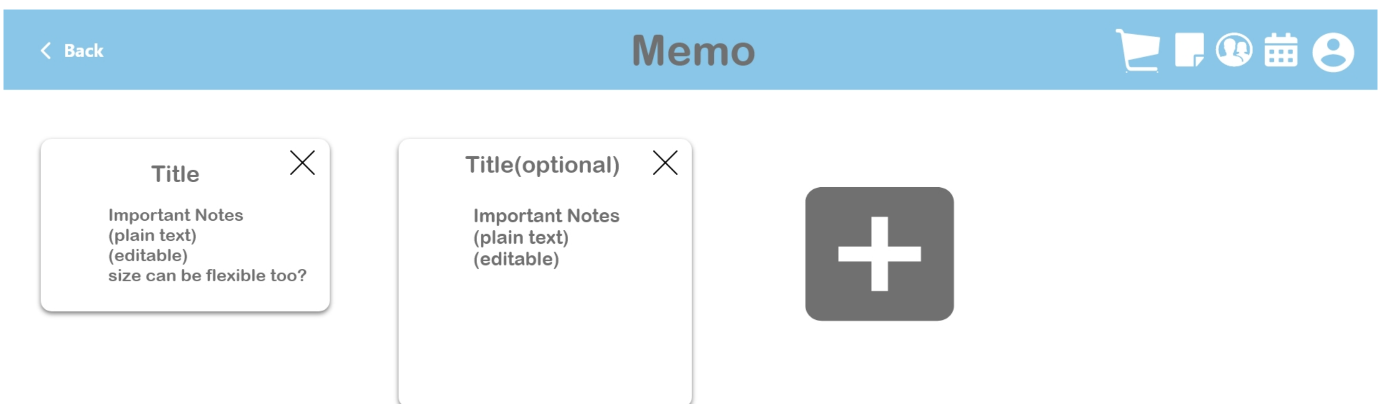

Back
Edit Profile
Given Name
Family Name
New Password
Confirm New Password
Your password must contain atleast one uppercase letter, one numerical character and one special character
Update
The Dream Team PTY

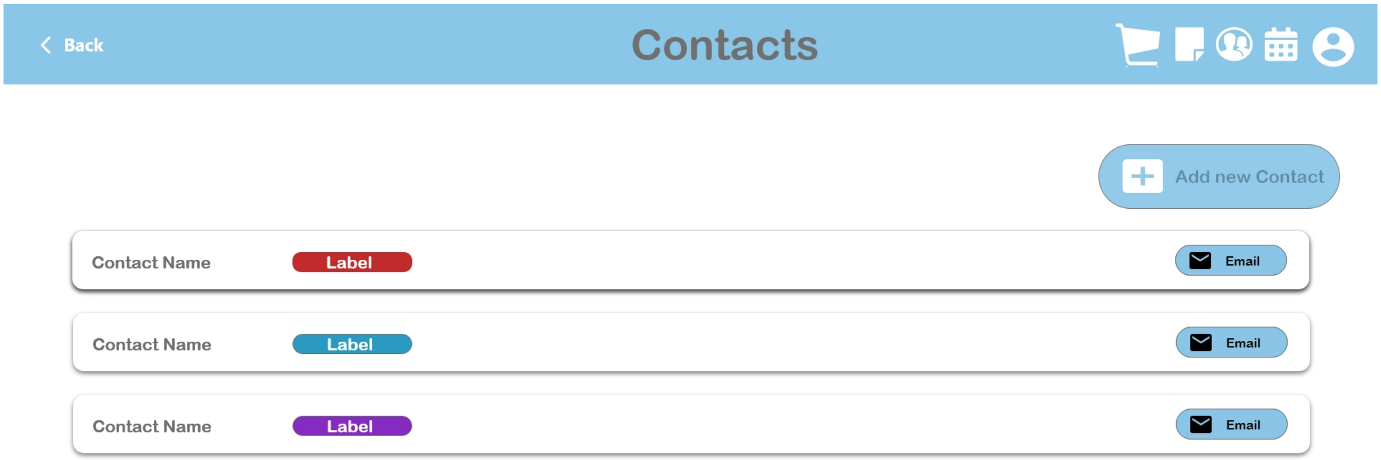

Back
Meetings
Upcoming
Past
Add new Meeting
Contact Name
Meeting Time and Date
Title
Agenda
View/Edit Note
Mark as Past
Delete

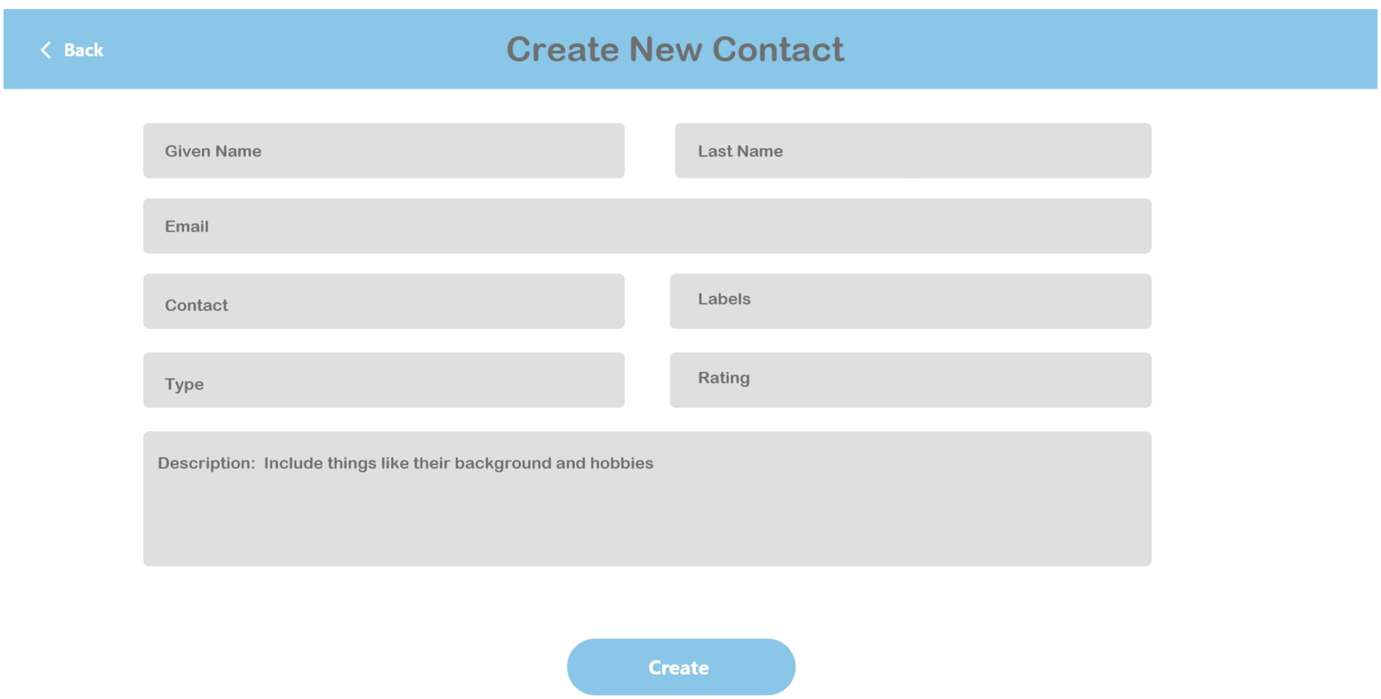

Back
Meetings
Upcoming
Past
Add new Meeting
Meeting Title
Contact N
ew/Edit Note
Mark as Past
Delete
Type Notes Here .


Add New Meeting
Title
Agenda
Date
Time
Notes ???
Add
The Dream Team PTY

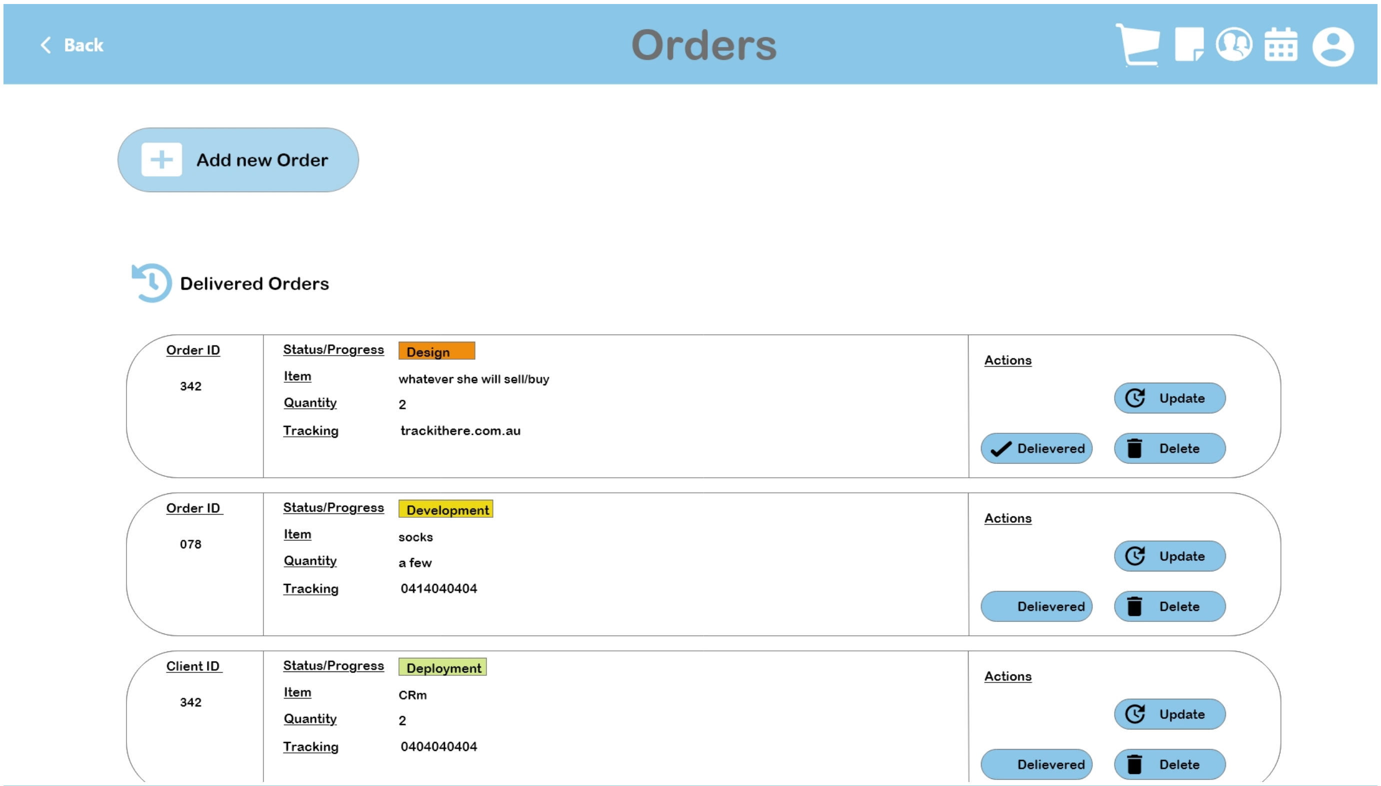

Edit Meeting
Title
Agenda
Date
Time
Notes ???
Cancel
Update
The Dream Team PTY


Back
Memo
Title
Important Notes
(plain text)
(editable)
size can be flexible too?
Title(optional)
Important Notes
(plain text)
(editable)

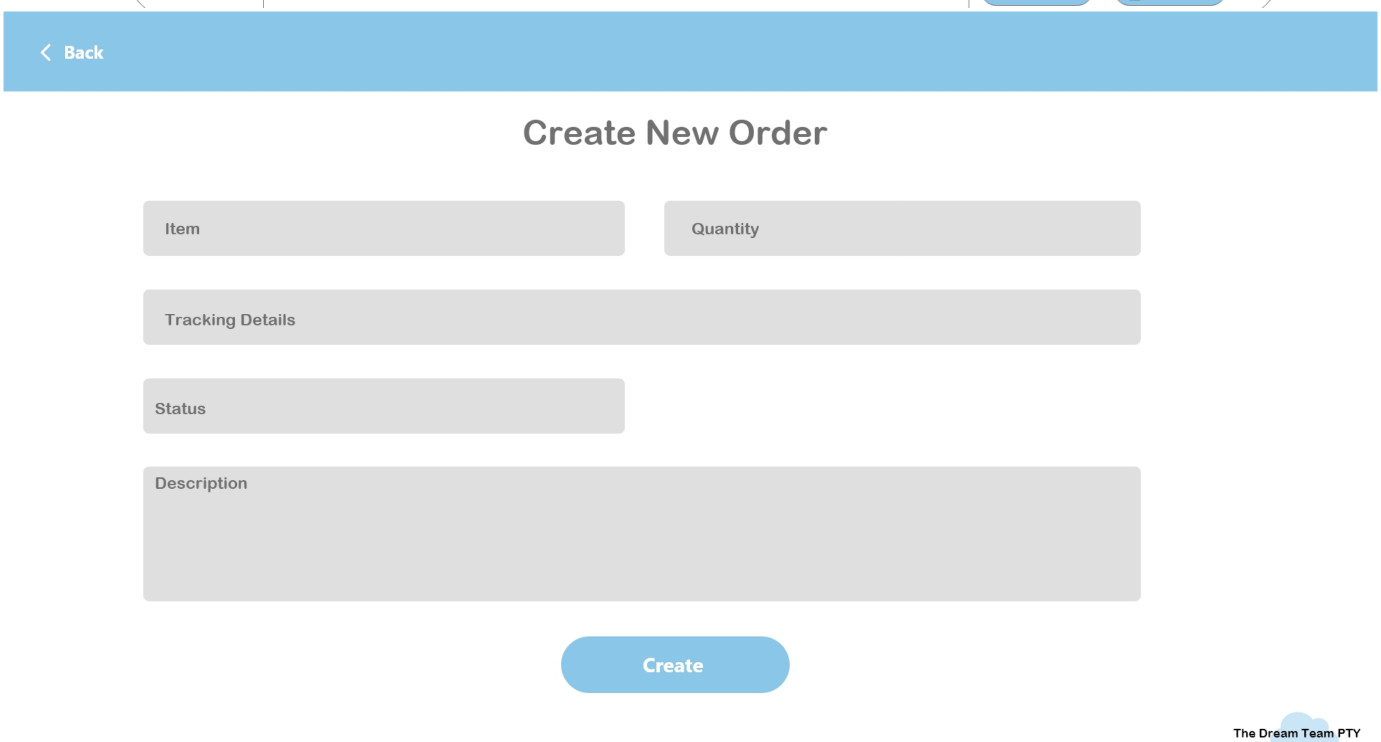




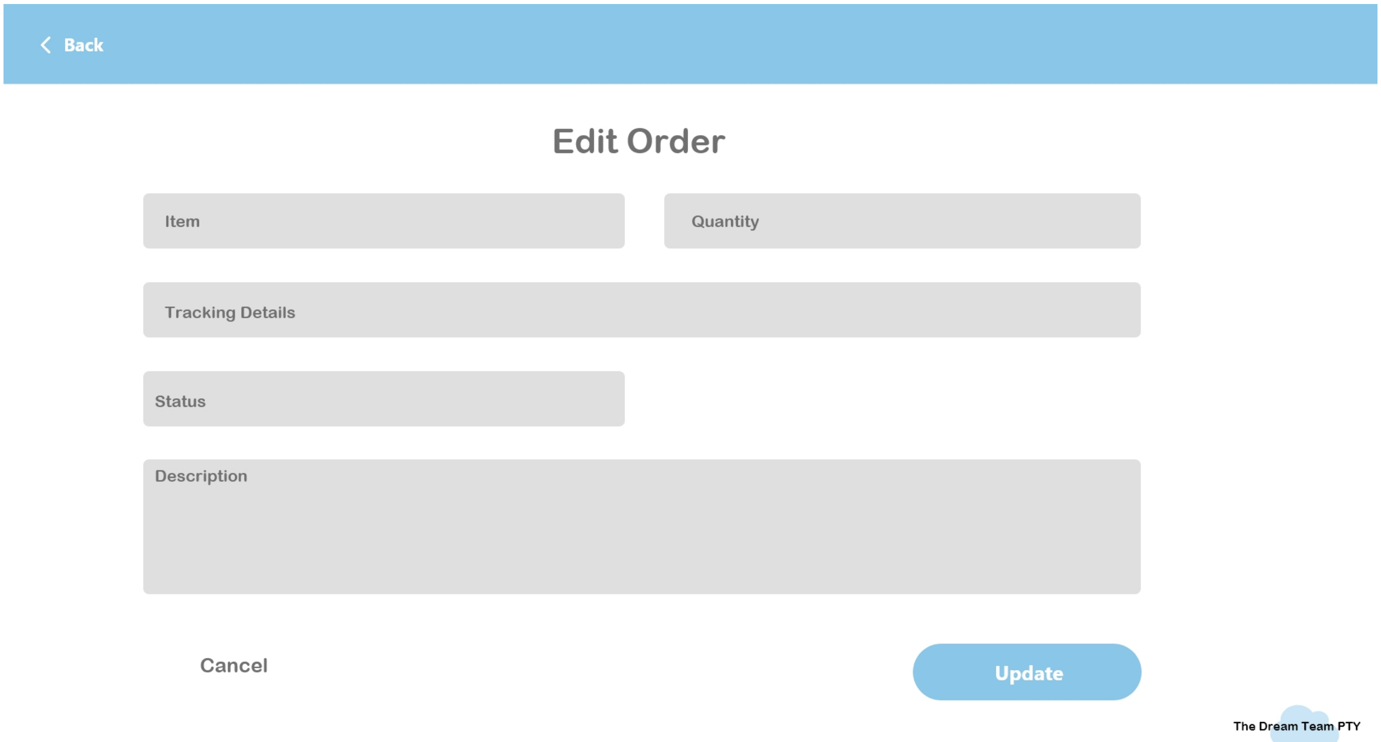
The Dream Team PTY



Back
Orders
Add new Order
Delivered Orders
Order ID
342
Status/Progress
Design
Item
whatever she will sell/buy
Quantity
2
Tracking
trackithere.com.au
Actions
Update
Delievered
Delete
Order ID
078
Status/Progress
Development
Item
socks
Quantity
a few
Tracking
0414040404
Actions
Update
Delievered
Delete
Client ID
342
Status/Progress
Deployment
Item
CRm
Quantity
2
Tracking
0404040404
Actions
Update
Delievered
Delete



Back
Create New Order
Item
Quantity
Tracking Details
Status
Description
Create
The Dream Team PTY



Tash comments:

Need to add back "delete contact" ability
Make order action section look more symmetrical: I don't like the one next to two action options
Maybe check the extent to which order info is captured by the system- don't want to show functionality that we aren't going to implement
If we have "edit profile" they should be able to edit all attributes - job roles may change etc

Checklist

Inception Checklist

Team Name: The Dream Team (Team 56)

Supervisor: Abhisha Nirmalathas

Process

· **Do you have a communication tool? If so, which one?**

Discord

· **Where are you storing team documents?**

Confluence - Home

· **Are there minutes of meetings? If so, give a link to where they are stored.**

Confluence - Meeting notes

Client Meetings

· **Have you set up a repository? If so, which one?**

Github - https://github.com/ccharansingh/DreamTeam

· **Evidence of tasks being assigned?**

Trello - https://trello.com/b/rQqkfFbF/sprint-1

· **Are all team members contributing? Elaborate as needed.**

Yes, all team members are contributing. We meet at least once a week as a team to discuss the progress of tasks and assign out any tasks that can be completed in the given week. We also meet in small groups when possible to pair program and talk through any issues we may be facing. All team members have contributed equally to the project and have set roles. However, no team member has taken these roles as being finite and have helped out with various tasks where possible.

Artefacts

What requirements artefacts have been developed? Give a link to them.

· Use Cases

Use Cases

. User Stories:

User Stories

· User Persona:

User Persona

· Motivational Model:

Motivational Model

What design artefacts have been developed? Give a link to them.

· Domain Model:

Domain Model - #Iteration 1

Domain Model #Iteration 2

· Design Class Model:

Design Class Diagram #Iteration 1

Design Class Diagram #Iteration 2

· Database/Schema Model:

Database Models/Schemas

https://dreamteam56.atlassian.net/wiki/spaces/DT/pages/15007752/iteration+2

. Front end UI Design:

Latest Iteration

Sprint 1 Checklist

Team Name: The Dream Team (Team 56)

Supervisor: Abhisha Nirmalathas

Process

· **Communication tool up to date?**

Yes, all communication is through Discord and the tool is being used regularly to update and communicate with team members and supervisor.

· **Team documents up to date, including meeting minutes, task assignments?**

Yes, Confluence page has been maintained and updated with meeting links and task assignments. Trello has been regularly utilised for task assignment.

Confluence Pages:

Supervisor Check Ins

Standups

Client Meetings

Trello Kanban Board:

· **Please give a link to your repository.**

GitHub Repository:

https://github.com/ccharansingh/DreamTeam

· **Are all team members contributing as equally as feasible? Elaborate as needed.**

All team members have contributed equally to the sprint. All team members attended all meetings held during the sprint and were always ready to offer help whenever required in another area. All team members continuously update each other on the status of their progress and any blocks in completing their allocated tasks.

· **Have you done a sprint retrospective?**

Yes, sprint retrospectives have been completed.

Retrospective - Development Sprint 1

Artefacts

Have requirements been updated?

Yes there have been updates to the requirement. As a team, we started our development sprints slightly earlier compared to the guideline timetable set by the teaching team. As a result, we have also conducted a sprint review with the client and received feedback on the deliverables that we delivered in our first development sprint.

User Stories

Have design artefacts been modified?

Motivational model and domain model changed based on feedback from supervisor after inception sprint.

Motivational Model

Domain Model - Iteration #3

What tasks were completed in the sprint?

The following deliverables were completed in the first sprint:

- Sign Up Page - for new users to create an account and be able to access the CRM
- Login Page - for users to be able to securely access the CRM system using their personal accounts
- Contacts Page - for users to be able to add new contacts and edit/delete existing contacts
- Notes Page - for users to create memo notes and edit/delete any existing notes

Do you have coding standards? If so, give link.

Yes coding and commenting standards have been decided on as a team. They were decided in a stand-up and all standards were agreed upon unanimously by the team.

Links:

Style and Conventions

Commenting Conventions

Do you have a testing plan? If so, give link.

Yes a testing plan has been established.

Testing Plan - Development Sprint 1

Do you have a deployment pipeline? If so, elaborate on your tool and status of deployment.

We have used GitHub actions to set a pipeline for our main branch so every time we push or pull request to the branch it automatically builds the react app and NodeJS and runs our test units and upon success then it deploys the built app to Heroku.

Extra Artefacts/Pages for Sprint:

Development Sprint 1 Report

Development Sprint 1 - Sprint Review

Sprint 2 Checklist

Team Name: The Dream Team (Team 56)

Supervisor: Abhisha Nirmalathas

Process

- **Communication tool up to date?**

Yes, all communication is through Discord and the tool is being used regularly to update and communicate with team members and supervisor.

- **Team documents up to date, including meeting minutes, task assignments?**

Yes, Confluence page has been maintained and updated with meeting links and task assignments. Trello has been regularly utilised for task assignment.

Confluence Pages:

Supervisor Check Ins

Standups

Client Meetings

Trello:

- **Please give a link to the repository.**

https://github.com/ccharansingh/DreamTeam

- **Are all team members contributing as equally as feasible? Elaborate as needed.**

All team members have contributed equally to the sprint. All team members attended all meetings held during the sprint and were always ready to offer help whenever required in another area. All team members continuously update each other on the status of their progress and any blocks in completing their allocated tasks.

- **Have you done a sprint retrospective?**

Yes, a sprint retrospective has been completed.

Retrospective-development sprint 2

Artefacts

- **What requirements have been completed?**

We have completed orders, meeting, email and profile page in this sprint. In the previous sprint, we completed the contacts, notes, user log in and sign up pages. Therefore, all high and medium priority requirements have been met as per the user stories.

User Stories

- **Have design artefacts changed since Sprint 1?**

No changes to design artefacts have been made since Sprint 1.

- **What tasks were completed in the sprint?**

We have completed orders, meeting, email and profile page in this sprint. We also did further testing for integration and unit tests as well as adding requirements to the contact and notes pages as asked by the client.

- **Is coding up-to-date?**

Coding is up to date. All coding standards have been followed as set out by the team. Code on GitHub is up-to-date as per standards set out below.

Links:

Style and Conventions

Commenting Conventions

- **Do you have test results? If so, give link.**

Yes, we have completed integration, unit tests and end to end tests. All the standards and results as set out below.

Links:

Quality Assurance

- **What is the deployment status?**

The app has successfully been deployed on Heroku using GitHub actions. You can see further information regarding this on the deployment confluence page.

Links:

https://dream-team-crm.herokuapp.com/

Quality Assurance

- **Do you have a handover date?**

Handover date has been established to be sometime between 25/10/21 and 29/10/21. The exact date and time is to be confirmed once we have our client, Yiran’s availabilities.

Extra Pages:

Development Sprint 2 - Sprint Review

Development Sprint 2 - Sprint Report

Product Checklist

Has the project been deployed? If so, please provide link

Project has been deployed - https://dream-team-crm.herokuapp.com/

Has the project been handed over to the client? **Yes**/no

Project has been handed over. The following items have been handed over:

- Client has been added to GitHub Repository - for access
- Exported confluence site for all documentation and minutes
- Licensing agreement confirming transfer of intellectual property rights
- User manual briefly showing the features of the site and what each function does
- Avatars - used for presentation - Client wanted access to this

Handover Details:

Date: 2/11/2021

Time: 3:00 pm AEDT

Attendees: Client (Yiran), Anagha, Chirag, Koosha, Natasha, Olivia

Further minutes and details: *link*

Overall Checklist

The first part is stating how and when the handover has happened.

You need to say when the handover meeting happened, if and where the project is deployed, and what was handed over,

Project has been handed over. The following items have been handed over:

- Client has been added to GitHub Repository - for access
- Exported confluence site for all documentation and minutes
- Licensing agreement confirming transfer of intellectual property rights
- User manual briefly showing the features of the site and what each function does
- Avatars - used for presentation - Client wanted access to this

Handover Details:

Date: 2/11/2021

Time: 3:00 pm AEDT

Attendees: Client (Yiran), Anagha, Chirag, Koosha, Natasha, Olivia

Further minutes and details: *link*

The second part is a 'boasting page', which says which aspects of your project the team is most proud of.

transfer data from Features To Check Out (Boasting Page)

The overall submission should be at most two pages.

Formatting

Sprint Review - Format

Background:

Date:

Attendees:

Agenda:

Meeting Minutes:

Refer to link below for in-depth meeting minutes and the recording of meeting.

Client Meetings

Client Response/Feedback:

Commendations:

Recommendations:

Sprint Report Format

Dates	
Contributions by team member	
Sprint achievements	
Backlog items, status	
Backlog items from previous sprint	
Backlog items removed from sprint	
Link to kanban board	
Link to retrospective	

Personal Development

Format

Heading <Title of learning resource>

Description <brief outline of resource>

Tools <tools required for tutorial>

Link/Access <way to access the tutorial>

Front End PD

Passing data and functions between class based components in react

How to Pass Data between React Components | Pluralsight

Project example in src/components/contact/contactList passing data from ContactList component to contactLabel component.

Back End PD

Full Stack Dev PD

MERN stack in 2 hours: https://www.youtube.com/watch?v=7CqJlxBYj-M

ARTICLE VERSION OF ABOVE : Learn the MERN stack by building an exercise tracker — MERN Tutorial | by Beau Carnes | Medium

Useful Links

Making views look good-without the work

Material-UI: A popular React UI framework

Great documentation on using react found on react's website

Forms – React (reactjs.org)

MERN stack youtube video

Learn the MERN Stack - Full Tutorial (MongoDB, Express, React, Node.js) - YouTube

MERN stack git rep-super easy to look at

Learn the MERN stack by building an exercise tracker — MERN Tutorial | by Beau Carnes | Medium

Nice to Have Features

- Deleting past meetings over a certain amount of time ago
- Having a label sort of thing for Progress of orders
- Password Requirements (e.g. must have 8 characters etc)

Sprint Planning

Retrospectives

Error rendering macro 'create-from-template' : Failed to find net.sf.hibernate.Session from the current thread

Title	Date	Participants
Retrospective - Development Sprint 1	09 Sep 2021	@ Natasha Ireland @ Olivia Ryan @ Anagha Giri @ kowji @ Chirag Charan Singh
Retrospective - Inception Sprint	09 Sep 2021	@ Natasha Ireland @ Olivia Ryan @ Anagha Giri @ kowji @ Chirag Charan Singh

Reports

Features To Check Out (Boasting Page)

Contact Labelling and Searching

Do you hate having to scroll through an entire list of contacts just to find the right one? Do you get frustrated by not being able to organise contacts into logical groups?

Not anymore!

The DreamTeam CRM provides users with the ability to search contacts by first OR even last name.

But wait…there's more!

Users can also create labels with a customised title and colour to help organise their contacts. They can use these labels to filter through all the contacts and view only those who have the selected label.

And it doesn't stop there!

Contacts can have not one, not two, not three but up to six labels making grouping and viewing that much easier.