This documents outlines the generation of the product backlog of the team's project.

User stories:

The table below was extracted from our 'Requirement Elicitation' document. The user stories identified were used as inspiration to create the product backlog: a list of tasks for the team to implement towards the completion of the project.

	As a [role]	I want (to) [do]	So I can [achieve]	Priority (HIGH / MEDIUM / LOW)
1	User	Register	Do anything else	HIGH
2	User	Login	See all my business card	HIGH
3	User	Create Business card	Have more business card	HIGH
4	User	Edit a Business card	Change a business card	HIGH
5	User	Delete a business card	Clear out old contacts	HIGH
6	User	See more information about a card	To see more information about the card, such as notes	HIGH
7	User	Keyword Search	Find a business card quickly	HIGH
8	User	Add more fields for a business card	have other info on a business card	MEDIUM
9	User	Add long notes to my business card	have other info about the business card	MEDIUM
10	User	Add tags to card	To be able to filter search by tags	HIGH
11	User	Filter Search by tags	To be able to filter search by tags	HIGH
12	User	Add existing tags	For the tag to be added to some business card in the future	LOW
13	User	Delete existing tags	Remove used tags	LOW
14	User	Edit existing tags	Change the name and any other info about a tag.	LOW
15	User	Search in various fields	To search more efficiently	LOW
16	User	Favorite/pin business card(s)	Keep track of frequently used business cards	LOW

Each user story fell into one of three 'Epics': card management, account management, and tag management.

Software Architecture:

The software architecture consisted of a website front-end and a server back-end, as outlined in the 'SystemComponent' artefact. Completing each user story then involved implementing endpoints in the back-end as well as appropriate user interfaces in order for the stories to be realised.

Product Backlog:

The table below highlighted some key tasks that made up the product backlog. Some tasks were created in order to facilitate other tasks such as T1 and T9.

Task Number	Task Name	Description	Relevant User Story(s)
T1	Code infrastructure and boilerplate	Establishing the software environment such as folder structure, database cluster connection, testing and CI, and dependencies.	N/A
T2	Creating a registration and login view	Creation of user interfaces for user login / registration.	1, 2
T3	Implementing user login endpoint	Implement the logic for the server to authenticate a user.	2
T4	Implementing user registration endpoint	Implement the logic for the server to authenticate a user.	1
T5	Creating card detail panel component	The panel will allow users to see a card's details, offers a form for them to edit or delete the card. This form will be recycled for card creation.	3, 4, 5, 6, 8, 9, 10, 16
T6	Implementing card creation (PUT) endpoint	Implement the logic on the server to accept details to place a new card record in the database.	3, 8, 9, 10, 16
T7	Implementing card update (PATCH) endpoint	Implement the logic on the server to accept a card's new details for updating its record in the database.	4, 8, 9, 10, 16
T8	Implementing card removal (DELETE) endpoint	Implement the logic on the server to receive a card's identification for removal from the database.	5
T9	Implementing authentication middleware	Implement middleware on the server that identifies users through authentication.	All except item user story 2
T10	Implementing image retrieval (GET) endpoint	Creation of a route for the application to retrieve images from the database with ease.	6
T11	Implementing user data retrieval (GET) endpoint	Endpoint allows the back-end to retrieve a user's information, including their owned cards and tags.	2
T12	Creating the home screen view	This view includes the user's stored cards . The cards displayed can be filtered via the search functionality or organised into favourites.	2, 7, 11, 15
T13	Creating the application's header component	The component includes navigation features and the search bar for users to access search functionality.	7, 11, 15
T14	Implementing tag creation (PUT) endpoint	The endpoint allows the server to save a tag associated with its user in the database.	12
T15	Implementing tag update (PATCH) endpoint	The endpoint allows the server to receive a tag's new information to update its record in the database.	14
T16	Implementing tag removal (DELETE) endpoint	The endpoint allows the server to remove a specified tag from a specified user.	13