# Overview

This is a two part project where there will be a public facing web app where there is information about the product.

## Sales Site

The sales site will be able to accept payments using the stripe api to sign a user up to a monthly subscription. As this site will offer plans it must also be able to display the information about what plans are on offer. Additionally, there should be an option for a custom plan for enterprise customers. Normal customers will not get this as they could make plans that are too low in cost to get any money and could cause slow downs as each individual account could have its own subscription. Thirdly, as we are interacting with normies we will also need to have a working support ticket system. This will allow logged in users to get help with their account but if they can’t login they can put in some to be decided information to allow them to create a ticket then use that to come back and check on the status of their ticket. In general this site must look professional and could have 3D icons and images like in microsoft and work on both mobile and desktop platforms and work in multiple browsers. Finally, it should fully load in less than 2 seconds

In short the site must (Current Points: 10):

* Accept Payments
* Charge users on a monthly basis
* Display information about the plans on offer
* Allow enterprise customers to make a plan to suit them
* Customer Support tickets for Logged in users
* Customer support for logged out (guest) users
* Professional layout and aesthetics use 3d renders like on bing
* Mobile and Desktop compatibility
* Work in multiple browsers
* Load in less than 2 seconds

## Web App

This web application is what the user will pay for access to. Once logged in, the account paying for the service is the admin/owner of the organization. In their dashboard they will be able to create users with a username and password. This password will then be able to be changed by the user to prevent the admin from logging into their account. Two factor (email) authentication or and authenticator app should be an option but isn’t required for the initial build. The administrator should be able to create individual accounts manually by entering a user name and password and be able to create one by importing a csv file. The service should also be able to securely generate passwords. These logins can only be viewed until the user logs in and changes the password. These users should be able to managed into groups where they can only see the messages that the other people in the groups share. A user should be able to have access to multiple groups and have a different feed for each group but a single feed.

In the main interface of the app, once a user is in an organization there should be a feed similar to twitter, here they have the main feed that is curated for what they need to know about, then there is a search for people to find things that have already been asked or find previous conversations. There is another tab for them to see their account, this will contain who they are, role in the company, profile picture, their email address and maybe a little bit about them. This is where they can view what they have sent. In the sending interface, there will be a title (similar to the subject of an email), content (where they write what they want to communicate)

//Make this wall of text into paragraphs with each section i.e Login, account creation, communication, direct messages.

## Infrastructure

Hosted on a cloud service with a free tier to get started with.

# Sales Site

## Layout

* Homepage –
* Pricing –
* Features – Short bitesize points: Internal Knowledgebase, AI enhanced search, intelligent ranking o
* Support –
* Sign Up –
* Sign In –

## Navigation

# Digital Product

Be similar to stack overflow or twitter in the way there is a feed and stack overflow systems like being able to say a best solouion

# Infrastructure