

Final Year Project: Mid-Term Delivery

StockCompete

Aaron Nolan: 18423054
Ben Strickland: 18704489

Initial Business Model summary	4
1.1 Project Concept	4
1.1.1 Company Description Summary	4
1.1.2 The Problem	4
1.1.3 Analysis of the Problem	4
1.1.4 Our Solution	4
1.2 Value Proposition Canvas [Figure 6]	5
1.2.1 Customer Profile	5
1.2.2 Value Propositions	5
1.3 Business Model Canvas	6
1.4 Market Analysis	8
1.4.1 Total Available Market (TAM)	8
1.4.2 Serviceable Available Market (SAM)	8
1.4.3 Target Market (TM)	8
1.5 Competitor Analysis [Figure 11]	8
1.5.1 Indirect Competitor: RobinHood	8
1.5.2 Direct Competitor: eToro	9
1.6 Financial Plan [Figure 12]	9
1.6.1 Expenses	9
1.6.2 Operating Revenue	9
1.6.3 Investment Income	9
2. Summary Functional Specification & Technical Description	10
2.1 General Description	10
2.1.1 Product/System Function	10
2.1.2 User Characteristics and Objectives	10
2.1.3 Operational Scenarios	10
2.1.4 Constraints	10
2.2 Initial Functional Requirements	11
2.2.1 Registration	11
2.2.2 Login	12
2.2.3 Buy	12
2.2.4 Sell	12
2.2.5 Add Funds	12
2.2.6 Remove Funds	13
2.2.7 Database [Figure 13]	13
2.2.8 Monthly Leaderboard	13
2.3 Initial Software Architecture	14
2.3.1 User	14
2.3.2 NodeJS Frontend (127.0.0.1:3000)	14
2.3.3 Django REST Framework (DRF) (127.0.0.1:8000)	15
2.3.4 Django Backend (Database) [Figure 13]	15
2.3.5 Alpha Vantage API Service	15

3. Project Timeline	16
3.1 Organisation and Business Case Timeline	16
3.2 Technical Specification with Prototype Development Timeline	17
References	17
Appendix	18
Figure 6 - Value Proposition Diagram	19
Figure 7 - Survey: Expectations on a stock web app	19
Figure 8 - Survey: Interest in gamification	20
Figure 9 - Survey: If yes, difficulties faced	20
Figure 10 - Survey: If no, reasons not	21
Figure 11 - Competitor Analysis Table	21
Figure 12 - Financial Plan (First Year)	22
Figure 13 - Database (ERD)	22

1. Initial Business Model summary

1.1 Project Concept

1.1.1 Company Description Summary

The name of our company is StockCompete and is based in Dublin, Ireland. Our aim as a business is to make stock trading as accessible to users as possible, regardless of their level of expertise. Our company provides a service of a stock trading web application hosting enticing and interesting features such as a User Dashboard, Monthly Leaderboards and Simplified Trading Information.

The initial target market for our service is non-professional traders based in the EU Market. From research, it has been clear that there is a higher demand for this type of service in the age range of 18-24, however this service will be available to all over the age of 18. We will in the future be looking to expand both outside the EU Market. The competitors to our target market host their own range of features with their service, however, they do not include a gamified UI, no hidden fees and simplified trading information. We also have a very low minimum account balance for our users.

1.1.2 The Problem

Many people who have displayed interest in entering the stock market have been put off from doing so, due to numerous factors such as the amount of knowledge needed to trade successfully, the cost of starting and the amount of time it consumes. For example, from a survey we conducted, 46.2% of participants said they did not begin trading because it was difficult to trade, and 23.1% of participants gave cost as a reason for not trading [\[Figure 10\]](#).

1.1.3 Analysis of the Problem

Having conducted surveys on the target market to identify the problems detailed above, we then completed research on resolving the problems. This research consisted of searching other completed research on the same problems, collecting a number of possible solutions and identifying the solution that best suits our market.

1.1.4 Our Solution

Based on our research, an enticing method of learning for users is gamification. This can be seen in stock trading and other areas alike, for example, RobinHood, Duolingo and Khan Academy. Additionally, based on responses to our survey, it is a feature of interest to our market [\[Figure 8\]](#). Using this alongside low commission fees, simplified trading information and the low minimum account balance, we aim to resolve the recurring issues of cost, time and complexity.

1.2 Value Proposition Canvas [\[Figure 6\]](#)

To start our project we conducted surveys on our identified customer segment, non-professional traders. We then used the results [\[Figure 7-10\]](#) to create a value proposition canvas. This is used to identify how our service helps our customers solve a problem.

1.2.1 Customer Profile

Customer Jobs: To determine what the customer wants, we first need to determine what they are trying to do. For this, we identified two customer jobs - they are trying to trade stock and trying to learn how to trade well.

Gains: We then identify what the customer expects and would like to have. According to our primary research, customers expect Stock Information, a User Account Section and a way to visualise performance [\[Figure 7\]](#). 73.7% of survey participants also expressed interest in applications that use gamification [\[Figure 8\]](#).

Pains: Finally, we identify any issues they have to start or during trading. According to customers in our research that have traded stock before, some of the issues they came across were Cost (37.5%), Difficulty Interpreting Data (37.5%) and Lack of Knowledge (37.5%) [\[Figure 9\]](#). Meanwhile, customers who haven't traded stock before said that Cost (23.1%), Time (15.4%), Too Complicated (38.5%) and Difficulty In Learning (46.2%) were the main factors that prohibited them from starting [\[Figure 10\]](#).

1.2.2 Value Propositions

Gain Creators: Based on the outlined expectations in the customer profile we know that a user dashboard and stock information is a necessity. However, they want a way to visualise performance and use gamification. For this, we chose the use of a points and leaderboard system.

Pain Relievers: As our customers find it difficult to interpret data or find it too complicated, we plan to create an easy-to-use UI. We plan to overcome the cost pain by only charging low fees. We also plan to use simplified stock trading information which will make it less difficult to understand and less time-consuming.

Products and Services: Having assessed our customer segment and discovered what would relieve their pains and gains, we have decided to create an easy-to-use stock trading web application that features low commission fees, a leaderboard and points system, a user dashboard and simplified trading information.

1.3 Business Model Canvas

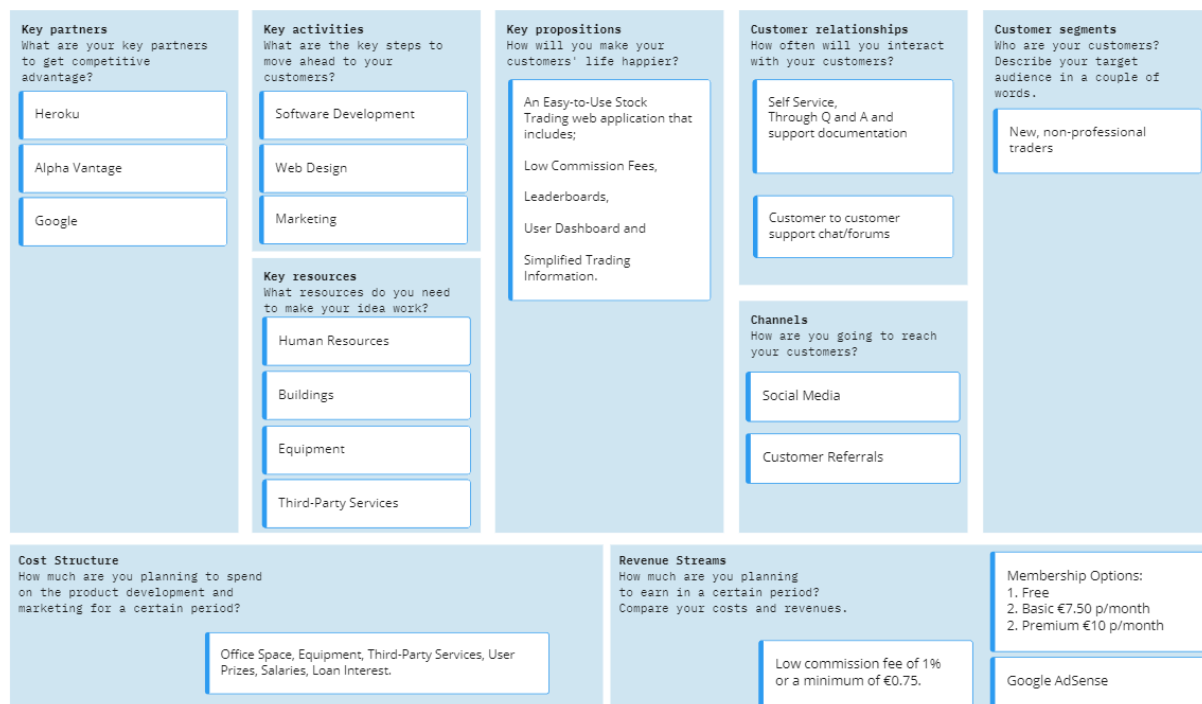


Figure 1: Above is our Business Model Canvas. This is used to define and communicate the business idea and are highlighted and expanded upon below.

Customer Segments: Our main customer segment consists of non-professional people new to trading. Although we have seen an increase of interest in trading in the age range of 18-24, “16 per cent of Britons aged between 18 and 24 began investing for the first time, compared with 10 per cent across all age groups.” [1], the service will be available to all ages 18 and above.

Value Propositions: Our value proposition (discussed in [Section 1.2](#)) is an easy-to-use stock trading application that features low commission fees, leaderboards, a user dashboard and simplified trading information. We determined that these features would have the greatest appeal to our target demographic based on our survey results - the cost of trading was a concern to participants [Figures 9 and 10], 46.2% of participants who had not previously traded found it too difficult to learn [Figure 10], and 68.4% of participants expected that a stock trading app would have its own user account section [Figure 7]. By including these features as part of our value proposition, these concerns will be mitigated, and people within our target market will be encouraged to use our app.

Customer Relationships: Our methods of maintaining customer relationships consist of a self-service Q&A section within the application with support documentation, and a customer-to-customer support forum. We chose these methods because, from our research, we found that providing a customer forum was an efficient method of fostering customer loyalty and a sense of community around the forum's associated product, allowing customers to address each other's problems - one report stated that “80% of community programs report an increase in asking and answering behaviours, which are critical to capturing implicit knowledge and making it transparent and accessible” [2].

Channels: Our channels for reaching customers are social media platforms and customer referrals. We chose these channels because studies conducted show that these are the most effective methods for reaching potential new customers and communicating with existing ones - surveys have shown that “*recommendations and reviews are trusted more than conventional advertisements*”, and “*customers that have been referred by a friend, family member, or colleague have a lifetime value that is, on average, 25% higher than that of other customers*” [3]. Additionally, social media provides an inexpensive way to advertise, and “*offer powerful targeting options so you can reach the right audience*” [4], meaning that based on user metrics e.g. demographic information, social media platforms will display ads for our platform to users who would best fit our target audience.

Revenue Streams: Our methods for generating revenue through the application consist of commission fees applied to customer transactions, (1% transaction fee, or a minimum of €0.75) Google AdSense integration that pays based on ad view count, and a tiered subscription service that offers benefits to customers who pay a monthly fee. This is further expanded upon in the financial plan in [Section 1.6](#).

Key Partners: Our key partners for the creation of the application are; **Heroku**, a cloud application platform that offers server hosting for the website to be accessible on a public domain and cloud storage for the website’s database, **Google**, which provides the AdSense program, allowing for an extra source of revenue to be generated by the website, and potentially driving customers to avail of our premium membership program, **Alpha Vantage**, which provides an API that delivers real-time information on a large number of stocks at a relatively low cost.

Key Activities: From our research [5], the key activities we will need to complete are; **UI/UX Design**, This process is to design, structure and style the application. It involves making an “*aesthetically pleasing*” layout for the user and taking into account the “*want, need, and feel*” of the user at every stage. **Backend Development**, Creating all the content of the server. This is “*the workings or backbone of the web app*”. It includes the “Framework”, “Server-Side Languages”, “Database”, “Web Server” and “Operating System”. **Frontend Development**, This allows the users to view and interact with the application. It is developed with “HTML”, “CSS” and “Javascript”.

Key Resources: The key resources necessary for the development of the application are; **Human Resources**, This will consist of software developers to program the functionality of the application, UI designers to create the layout and structure of the application and marketers to create advertising campaigns promoting the application, **Buildings**, Staff will require an office space to work in, and the company will need a headquarters with which outside parties can contact them and organise meetings, **Equipment**, This will include hardware such as computers and associated tools for employees to work on, **Third-Party Services**, This will include services necessary to integrate for the development of the application, such as Alpha Vantage’s API, Google AdSense program and Heroku’s cloud application platform.

Cost Structure: This is detailed in the financial plan in [Section 1.6](#).

1.4 Market Analysis

Market Analysis is a way of studying the market in a specific area, identifying key attributes and estimating population sizes to better understand the target demographic. This is then used to clarify aspects of the business idea to better suit the market demands.

1.4.1 Total Available Market (TAM)

The Total Available Market consists of everyone that has internet access and a bank account and aged 18 or older. From our research, we have found that:

- 66% of the world population is aged 18+ [\[6\]](#), 5.16 billion people
- 69% of people have a bank account [\[7\]](#), 3.54 billion people and
- 59.5% of people have internet access [\[8\]](#), 3.07 billion people.

With the assumption that people with internet access have a bank account, we calculate that 3.07 billion people (39% of the world population) are aged 18 or over, have internet access and have a bank account, thus forming our Total Available Market.

1.4.2 Serviceable Available Market (SAM)

From the initial launch of our service, we would be available globally but will only deal with trades in the European Market. The total trade completed by the EU is 14% [\[9\]](#). If we were to assume that all of the TAM trades or will trade, we can take 14% of this population to calculate the Serviceable Available Market as 429.8 Million people.

1.4.3 Target Market (TM)

Currently, eToro has a total of 23.2 Million users [\[10\]](#). These users aren't specific to our target market so based on our calculation from TAM, we will take 6.5% of 23.2 Million, 1.5 Million, as the number of users currently being served. This brings our Target Market to 428.3 Million people.

1.5 Competitor Analysis [\[Figure 11\]](#)

Competitor Analysis is a way of comparing a business idea to current market competitors and clearly identifying the key differences and selling points of their services to yours. Some of the top competitors our service would be up against are RobinHood (US) and eToro. We have identified these two platforms, as they both possess characteristics similar to what we plan on implementing into our application.

1.5.1 Indirect Competitor: RobinHood

Robinhood is a stock trading platform that utilises gamification as a feature of its service. It uses social media and advertising to publicise and market its products and generates revenue with a 'Payment for Order Flow' process. This allows them to offer a 0% commission fee to their users.

"Payment for order flow (PFOF) is the compensation and benefit a brokerage firm receives for directing orders to different parties for trade execution. The brokerage firm receives a small payment, usually fractions of a penny per share, as compensation for directing the order to a particular market maker." [\[11\]](#).

However, the EU is currently investigating this process and is looking to ban it from the European market entirely [\[12-13\]](#). It is currently banned in a number of countries in the world such as; United Kingdom, Canada and Australia [\[12\]](#).

1.5.2 Direct Competitor: eToro

eToro is a stock trading platform with a global market, excluding the US. eToro offers a 0% commission fee to its users and generates revenue from a variety of sources, primarily through what are considered 'hidden fees'. These fees are not apparent to the user until they are being charged. eToro has not implemented gamification into its platform.

1.6 Financial Plan [\[Figure 12\]](#)

The financial plan is a one-year projection of the cost structure (Expenses), as identified and discussed in [Section 1.3](#), the operating revenue and the investment income which are detailed below.

1.6.1 Expenses

Developer Salaries* [\[14-15\]](#): Two senior (€65,000 each p/year) and one junior (€30,000 p/year) developers create all the backend and some frontend logic for the application.

UI/UX Salaries* [\[16\]](#): Two Web Designers (€30,000) to create the style, layout and main frontend functionality.

Marketing Salaries* [\[17\]](#): A team of two marketers (€31000 each p/year) to market the application.

Heroku [\[18\]](#): Server hosting to make the application globally accessible.

Rent [\[19\]](#): Office space is required for employees to work (€1,000 p/m).

Equipment [\[20-24\]](#): This consists of various laptops, monitors, keyboards and mouses depending on the job.

Miscellaneous: This is an extra allocation for unexpected fees.

User Prizes: This is for the leaderboard system feature of the application.

Loan Interest [\[25\]](#): The BOI loan will be paid back in monthly instalments over the course of a year.

** Salaries are decreased as share options will be given to employees as equity to compensate [\[26\]](#).*

1.6.2 Operating Revenue

Commission Fees [\[13, 27\]](#): Based on eToro and Robinhood's statistics we aim to have 50,000 users over 6 months. We estimate each user will make on average 1 trade p/week.

Subscriptions: We aim to get 30% subscribed to monthly plans. 20% would be the basic plan and 10% would be the premium.

Google AdSense: Using Google AdSense and slowly reaching 50,000 views per month we would aim to generate €10,000 per year. This was calculated using the Google AdSense calculator.

1.6.3 Investment Income

Enterprise Ireland [\[28\]](#): We will be looking for a €150,000 investment.

BOI Loan [\[25\]](#): This is to cover extra costs, with an interest rate of 6.31% at a monthly repayment rate of €1,724.18.

2. Summary Functional Specification & Technical Description

2.1 General Description

To start creating a technical description of the business concept, we first need to clearly define the core functions of the service, the user characteristics (who the user is, what they like, what they don't like etc.) and basic operational scenarios.

2.1.1 Product/System Function

The core functionality of the system that we plan to implement are: Registration, Login, Buy, Sell, Add Funds, Remove Funds, Database. Each of these functions are discussed in more detail in [Section 2.2](#).

2.1.2 User Characteristics and Objectives

The application will be made publically accessible via the use of Heroku's servers, so anyone with an internet connection will be able to access it. It is intended to help users buy and sell stocks at a low fee cost and allow them to visualise the performance of their currently owned stock. The platform is catered to non-professional people that are new to trading, and have a minimal amount of knowledge in the field. It will have an easy to use UI, and the user will only require a minimal amount of technical knowledge.

2.1.3 Operational Scenarios

Upon accessing the application, the user will be brought to the homepage. From there, they will have the ability to search for a stock and choose one from the displayed results. Options for registering a new account or logging in with an existing one are available through links displayed in the navbar. On the page for a specific stock, data such as the current price is displayed, and options for buying and selling this stock are displayed if logged in, given that the user has enough money in their balance to buy, or has the stock in their portfolio to sell that has not been purchased within the same day.

Users will be required to register a new account or log in with an existing one before stock can be bought or sold. If they choose to register, they will have to provide their name, email address, phone number, PPS number, home address, date of birth and password. If they choose to log in, they will have to provide their registered email and password. If the provided details are incorrect, they will be prompted to re-enter, otherwise, they will be transferred to the homepage. Once logged in, the user dashboard will be accessible from any page of the application via a link on the navbar. On the dashboard, the user can view information on their currently owned stocks, such as the quantity owned of each. The user details provided at registration and the history of user transactions will also be displayed.

2.1.4 Constraints

Time: We are constrained by a tight deadline of 15th April 2022. We are developing the application while simultaneously dealing with commitments between college and external obligations, such as assignments, study, exams and graduate role interviews.

Accuracy: We are dependent on the accuracy of Alpha Vantage when displaying the stock graphs and need to define a way of fairly measuring user performance.

Stock Trading Knowledge: We have prior background knowledge on stock trading - however, not on a professional level, and we will continually expand our knowledge in the field as the project progresses.

Usability: The application must be as easy to use as possible so that it is accessible to our target demographic.

Express JS: We have prior experience using the Express JS language and have knowledge gaps that may need to be addressed as we progress through the project.

AlphaVantage API: We are limited to 5 calls per minute and 500 per day, this means we need to be strategic in how and where we make the calls so that we don't use up our calls too quickly.

2.2 Initial Functional Requirements

Having identified the main functions of the service in [Section 2.1.1](#) we will now discuss each in detail outlining their criticality, any challenges or technical issues and any dependencies it has on the system. Each of these functions will be created in the prototype and the functional layout will be as shown in the figure below.

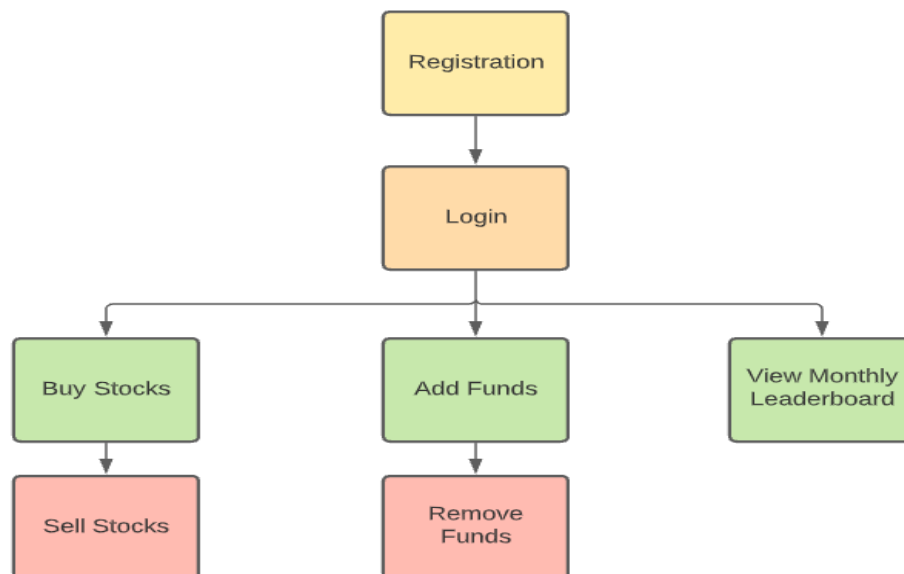


Figure 2: Functional Layout Diagram

2.2.1 Registration

Description: This function allows us to create a new user, with a list of specified data. The registration form will be available to access by clicking the link in the navbar. The form will collect the following information; name, email, password, phone number, PPS Number, address and date of birth.

Criticality: This function is critical to the application as almost all other functions will in some way be dependent on the data stored from this form. It helps us uniquely identify a user and in other functions link trading data, e.g. how many shares of Amazon they have bought, to a specific user.

Technical Issues: The form will be created in HTML and designed using Bootstrap. The function will be handled using asynchronous JavaScript which will create a user in the APIUser table of the MySQL Database in the Django Framework.

Dependencies: This function is not dependent on any other part of the system.

2.2.2 Login

Description: This will allow a user to gain access to their account and its associated information. The user will sign in to the form with their email and password which will be cross-referenced with the APIUser model for authentication. If successful they will be redirected to the homepage with a successful login message.

Criticality: This functionality is critical as it blocks access to certain parts of the site and allows us to monitor access to features such as the user dashboard, buy and sell.

Technical Issues: The system will have an error message occur in the instance a user's credentials are invalid.

Dependencies: This function depends on the completion of the registration process identified above.

2.2.3 Buy

Description: When the user is signed in, they will be able to purchase the stock of their choice. They will access this by going to an individual stock's page while signed in, filling out the buy form with the number of euros they wish to buy stock and selecting the buy by clicking the buy button. This will then create an entry in the purchase table of the Database.

Criticality: The function is a core component of the basic functionality of the application.

Technical Issues: The user will not be able to purchase a stock if they are not signed in, this will be verified via their auth token stored in browser storage.

Dependencies: This function is dependent on the login functionality identified above.

2.2.4 Sell

Description: When the user is signed in, they can sell any stock currently within their portfolio. This will be done by going to the page of the stock they wish to sell while signed in, inputting the quantity of stock they wish to sell (which is limited to the amount in their portfolio) into the sell form and confirm by clicking the sell button. This will then send a request to the backend, which will update the values for the user's balance and owned stock respectively in the MySQL database.

Criticality: This function is a core component of the basic functionality of the application.

Technical Issues: The user will have to be signed in (verified via auth token), have purchased stocks available in their account (checked in the portfolio table in the MySQL Database) and have not purchased those stocks on the same day (checked in the purchase table in the MySQL Database).

Dependencies: This function is dependent on the login and buy functionality.

2.2.5 Add Funds

Description: When the user is signed in, they will have the ability to add funds to their account balance. This will be done using buttons on the user dashboard page which, when clicked, each add preset amounts. (e.g. €50 or €100) This will then send a request to the backend, which will update the value for the user's balance.

Criticality: This function is a core component of the basic functionality of the application. The user will require funds in order to purchase stocks.

Technical Issues: The user will have to be signed in (verified via auth token) and use the selected amount of funds to add them to the account balance.

Dependencies: This function is dependent on the login functionality identified above.

2.2.6 Remove Funds

Description: This function will allow a user to remove funds from their account balance. The user can input a value into an input box of a figure less than or equal to their balance and press remove, this will update the balance of the account.

Criticality: This function allows the removal of funds from the account. In comparison to other features, this does not impact any other functionality but is a core component of the usability of this service.

Technical Issues: The user is signed in (Check auth token) and has a balance greater than 0 (Check balance amount) and input is less than or equal to the balance (Compare input to balance)

Dependencies: This function is dependent on the login (auth token) and the add funds functionality so that there is a balance to remove in the first place.

2.2.7 Database [\[Figure 13\]](#)

Description: This is situated in the Django Backend. It holds the following tables; APIUser, Account, Leaderboard, Stock, Purchase, StockBalance. These tables hold all the data that will be used in the application with the exception of the time-series data provided by Alpha Vantage.

Criticality: This database holds all the data used for the entire application. Most functionality will need to access data to function, making this the most critical point in the application.

Technical Issues: Accessibility through serializers for each table will need to be created for each table.

Dependencies: There are no dependencies for this function.

2.2.8 Monthly Leaderboard

Description: This is an automated functionality. When a user purchases a stock or sells a stock a number of points will be added to their account. This will then be compared to other users on a monthly leaderboard. The user will get to select their anonymous name which will be displayed on the leaderboard and allow them to visualise their performance to others. Users will then receive prizes if they are in the top 50 that month.

Criticality: This is an additional functionality but the unique selling point of the product. However, if broken it won't impact the rest of the website.

Technical Issues: The user is signed in (Check auth token), and their monthly score will be calculated based on their transaction list. This score will then be displayed on the leaderboard next to their name and current rank.

Dependencies: This function is dependent on the login (auth token) functionality and indirectly dependent on calculated scores based on the use of the buy and sell functionality.

2.3 Initial Software Architecture

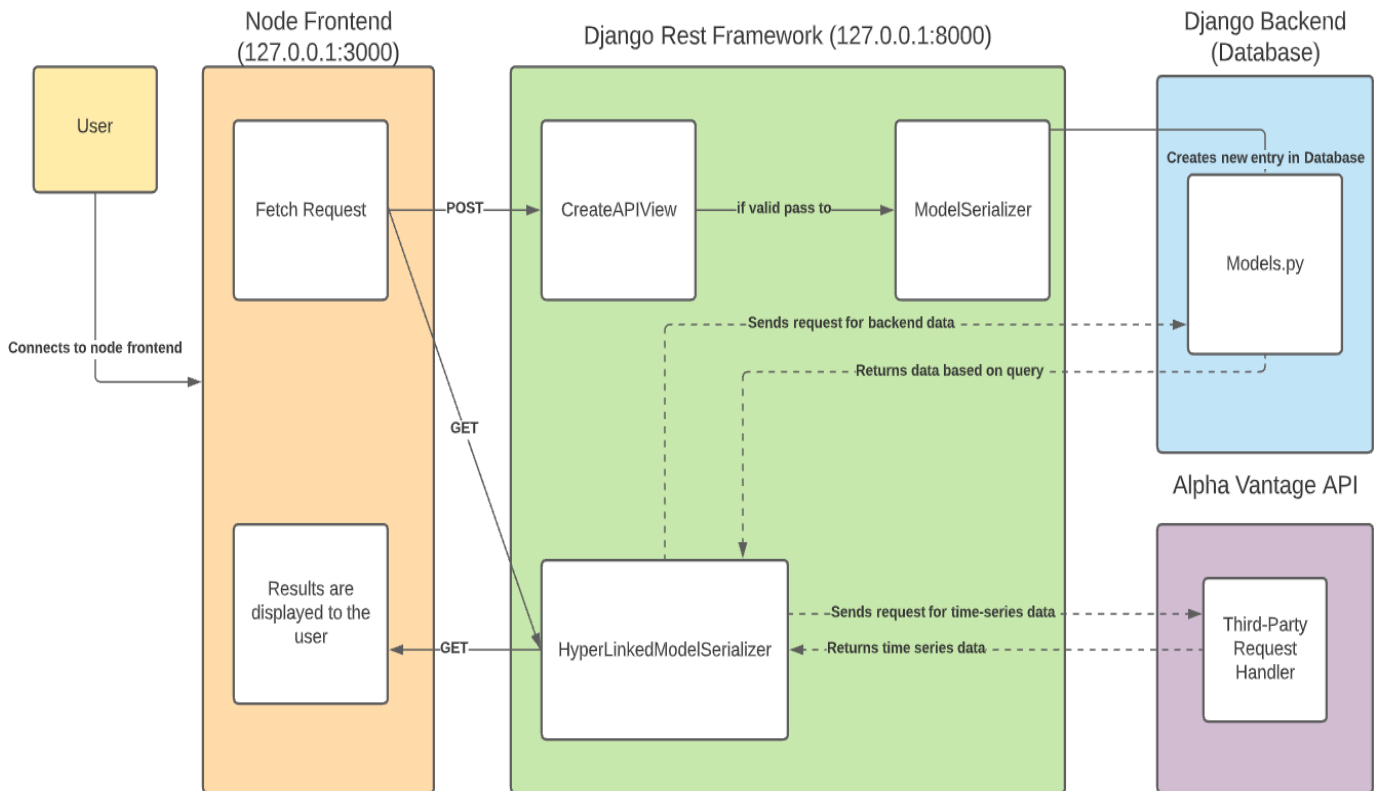


Figure 3: Software Architecture Diagram

The system architecture defines the high-level components of a system, how these components are structured and how they will interact with each other. For the system architecture of this system, we plan to use NodeJS as our frontend framework, Django REST Framework as our REST API and Django as our backend framework. We will also include a third-party API called Alpha Vantage to supply us with time-series stock data. With reference to the diagram above, the components will be detailed below:

2.3.1 User

The user is our customer. They will access the service via the web browser by directing it to the frontend URL (127.0.0.1:3000). The user will now be shown content by the NodeJS Frontend.

2.3.2 NodeJS Frontend (127.0.0.1:3000)

This component deals with displaying content to the user and is located at 127.0.0.1:3000. It allows the user to interact with the application. Depending on the user's interaction with Node, at some point, a fetch request will need to be created. This is the method by which the Node Framework interacts with the Django REST Framework (DRF). There are two main types of requests used, GET and POST. GET will retrieve information from DRF, which is then displayed to the User and POST which will send data from the User to DRF. The main challenge that needs to be handled with Node is Version Control. This will be JavaScript, Node, NPM and ChartJS as some or all of the service's features will stop working if these aren't compatible.

2.3.3 Django REST Framework (DRF) (127.0.0.1:8000)

As a REST API, DRF is the middle ground between the Node Frontend and either the Django Models or the third-party API service, Alpha Vantage. This API will read in GET and POST requests which have two separate processes. In the instance a GET request is received for data from a Django Model, with the option of authentication checks, DRF will send a query to the Model for specific data and return the result to the Node Frontend. A similar process is carried out for the Alpha Vantage data, however, the query is different. When DRF gets a POST request, the data is processed and in some cases, an authentication check is done before sending a create request to a specific Django Model. DRF has a lot of packages and configuration settings that will pose the biggest challenge when completing this section.

2.3.4 Django Backend (Database) [\[Figure 13\]](#)

The Django Backend or Database is a collection of database tables called Models that store data. They have specific attributes and types which are defined when a model is created. The database, as outlined in [Section 2.2.7](#), holds all the data in regards to the user, including their account, purchases, stock balance, the leaderboards and static stock info. This is used in every section of the application. Requests will come in from DRF and filter through the models for the desired results which are then returned to the DRF.

2.3.5 Alpha Vantage API Service

Alpha Vantage is a third-party service that provides a vast range of data in the context of trading. Due to the limitations of API calls on a free account, we will only be using the time-series data for individual stocks which will be displayed to the user on charts. A request will be sent from the frontend to the DRF and then to Alpha Vantage which will return the requested time-series data for a given stock. This returns through DRF and the frontend to be displayed in a chart, which is created using ChartJS. The only challenge with this process is handling the JSON object and ensuring the authentication process works.

3. Project Timeline

To ensure that we understand and complete all sections of the next steps of the project, we created a Gantt Chart to show the project timeline. This chart identifies the tasks that need to be completed, the length of time it will take to complete each task (in days), the expected start and finish dates and the person responsible for each task.

Due to the structure of the project and also the number of tasks needing to be completed we split the project chart into two sections, 'Organisation and Business Case' and 'Technical Specification with Prototype Development'. This shows the order in which the project will be done and ensures the graphs are readable on the page, as when combined on one image they weren't readable.

3.1 Organisation and Business Case Timeline



Figure 4: Gantt Chart of Business Documentation

The Organisation and Business Case Timeline breaks down into four smaller components that spread over 3 weeks from 10th Jan - 5th Feb. These four sections consist of the Business Case, Market Analysis, Research and Organisation Section.

The tasks Aaron will complete, as outlined in Figure 4, are Proposed Market, Value Propositions, User Segments, Key Resources, Cost Structure, Industry Trends, Competitor Profiles, Three-year financial plan, the Research section and the Management Report.

The tasks Ben will complete, as outlined in Figure 4, are the Business Case section, Revenue Streams, Key Activities, Pricing Model, Competitive Advantage, Three-year financial plan, the Research section and the Management Report.

3.2 Technical Specification with Prototype Development Timeline

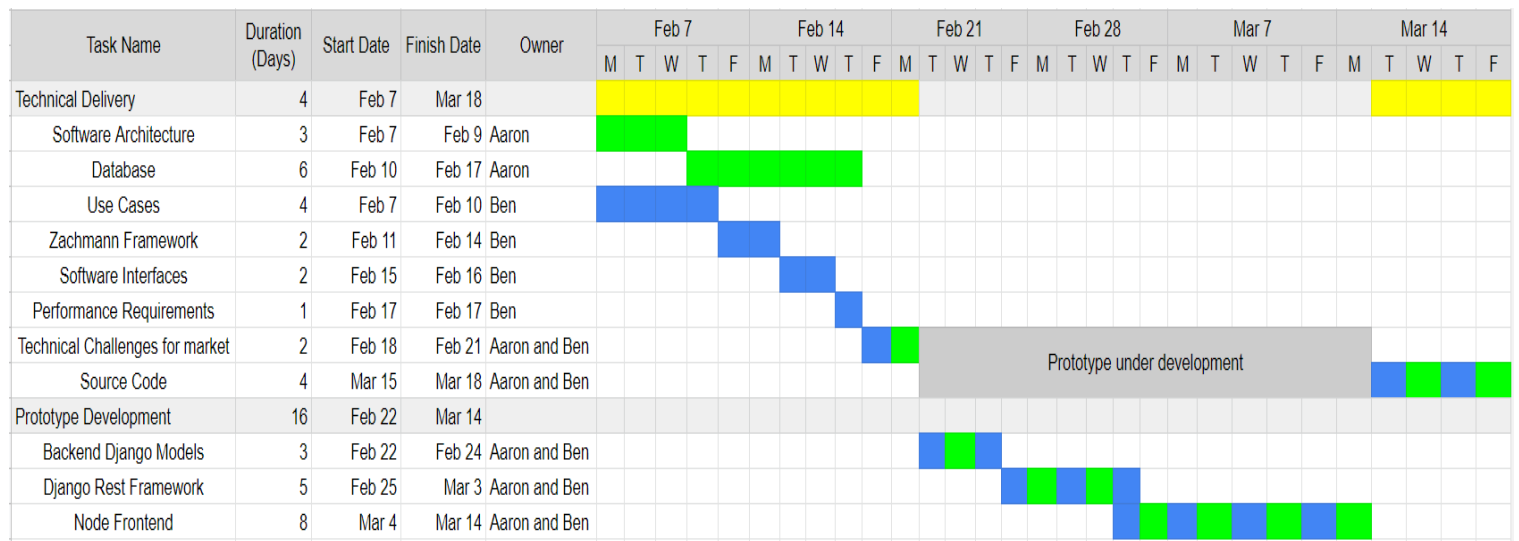


Figure 5: Gantt Chart of Technical Documentation and Prototype Development

The Technical Specification and Prototype Development Timeline continues on from the Organisation and Business Case Timeline and is carried out from 7th Feb - 14th March. It contains the breakdown of tasks associated with the technical section of the project, which are broken into two main sections - Technical Delivery, which focuses on the main topics of the report that cover the technical specification of the project, and Prototype Development, which focuses on the development of the application's prototype.

Although the timeline is not expanded until the deadline, there are other commitments that will also take priority throughout this timeline and will continuously push this project timeline forward. This is why we have a few weeks available in addition to this timeline.

The tasks Ben will complete, as outlined as Figure 5, are Use Cases, Zachmann Framework, Software Interfaces, Performance Requirements, Technical Challenges for market, Source Code, Backend Django Models, Django REST Framework and Node Frontend.

The tasks Aaron will complete, as outlined as Figure 5, are Software Architecture, Database, Technical Challenges for market, Source Code, Backend Django Models, Django REST Framework and Node Frontend.

References

1. Darbyshire, M. (2021, April 9). Gen Z embraces investing — with signs of caution. <https://www.ft.com/content/2ca31de6-2a2a-4e8f-8e0c-f5c313ce8fe3>
2. Schram, M. (2021, January 12). Transforming Your Customer Forum into a Customer Community. <https://www.cmnty.com/blog/transform-customer-forum/>
3. Miller, G. (n.d.). 14 Benefits of Referral Marketing to Inspire Your Strategy. <https://www.annexcloud.com/blog/14-benefits-referral-marketing-inspire-strategy/>
4. McLachlan, S. (2021, June 29). 22 Benefits of Social Media for Business. <https://blog.hootsuite.com/social-media-for-business/>
5. Lomas, A. (2021, September 13). A 15-Step Guide on How to Build a Web App. Retrieved from <https://www.netsolutions.com/insights/how-to-build-a-web-app/#designing-ux-ui-stage>
6. Ang, C. (2021, June 18). Visualizing the World's Population by Age Group. Visual Capitalist. <https://www.visualcapitalist.com/the-worlds-population-2020-by-age/>
7. Findex. (2017). Account Ownership. Retrieved from www.shorturl.at/lwzAY
8. Statista. (2021, September 10). Worldwide digital population as of January 2021. <https://www.statista.com/statistics/617136/digital-population-worldwide/>
9. Eurostat. (2021). International trade in goods - Statistics Explained. www.shorturl.at/fqsJT
10. Statista. (2022, January 11). Number of registered users of eToro from 2008 to Q2 2021. www.shorturl.at/klpvT
11. Carey, T. W. (2021, December 3). Payment for Order Flow (PFOF). Investopedia. <https://www.investopedia.com/terms/p/paymentfororderflow.asp>
12. Kolakowski, M. K. (2021, October 22). SEC Considers Banning Payment for Order Flow. www.shorturl.at/cyl27
13. The issue of payment for order flow. (2021, August 4). FESE. www.shorturl.at/nHLO7
14. Senior Software Engineer Salary in Ireland. (n.d.). PayScale. Retrieved December 9, 2021, from https://www.payscale.com/research/IE/Job=Senior_Software_Engineer/Salary
15. PayScale. (n.d.). Junior Software Engineer Salary in Ireland | PayScale. Retrieved January 2022, from https://www.payscale.com/research/IE/Job=Junior_Software_Engineer/Salary
16. Web Designer Salary in Ireland. (n.d.). PayScale. Retrieved July 31, 2021, from https://www.payscale.com/research/IE/Job=Web_Designer/Salary
17. Marketing Coordinator Salary in Ireland. (n.d.). PayScale. Retrieved January 13, 2022, from https://www.payscale.com/research/IE/Job=Marketing_Coordinator/Salary
18. Pricing. (n.d.). Heroku. Retrieved January 13, 2022, from <https://www.heroku.com/pricing>
19. Private offices for 8 - €1,000/month - Dublin. (n.d.). Rubberdesk. Retrieved January 13, 2022, from <https://rubberdesk.ie/office-space/for-8-in-dublin-just-1000-per-month/c75fd54a8b0c7>
20. ThinkPad X1 Carbon Gen 9. (n.d.). Lenovo Ireland. www.shorturl.at/mzA03
21. Yoga Slim 7 Pro Gen 6 (16" AMD). (n.d.). Lenovo. www.shorturl.at/ruEHJ
22. Yoga 6 (13" AMD). (n.d.). Lenovo. www.shorturl.at/aqMU2
23. SAMSUNG LS34J550 Quad HD 34" LED Monitor - Grey. Currys. www.shorturl.at/esPW1
24. Trust Primo Keyboard & Mouse. (n.d.). Harvey Norman. www.shorturl.at/ahxRV
25. Business Loans Benefits. (n.d.). Bank of Ireland. www.shorturl.at/gpyA9
26. How much employee equity should you give your first hires? (2021, November 5). Carta. <https://carta.com/blog/startup-employee-equity/>
27. Statista. (2021, December 13). Number of users of Robinhood 2014–2021. <https://www.statista.com/statistics/822176/number-of-users-robinhood/>
28. Have a Start-up Idea? (n.d.). Enterprise Ireland. <https://www.enterprise-ireland.com/en/funding-supports/Company/Have-a-Startup-Idea/>

Appendix

Figure 6 - Value Proposition Diagram



The Value Proposition Canvas visually represents the customer profile created from the conducted research on the customer segment and then displays the solutions to the pains and gains identified in the customer profile in the Value Proposition Canvas.

Figure 7 - Survey: Expectations on a stock web app

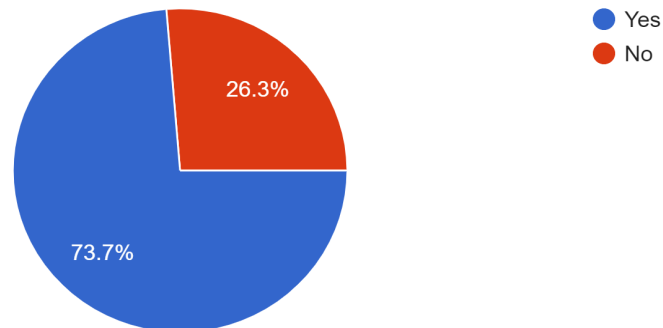


The survey responses to the question “What would you expect on a stock trading site?”.

Figure 8 - Survey: Interest in gamification

Do you enjoy using game style apps, such as Duolingo or Khan Academy?

19 responses



The survey responses to the question “Do you enjoy using game-style apps, such as Duolingo or Khan Academy?”.

Figure 9 - Survey: If yes, difficulties faced

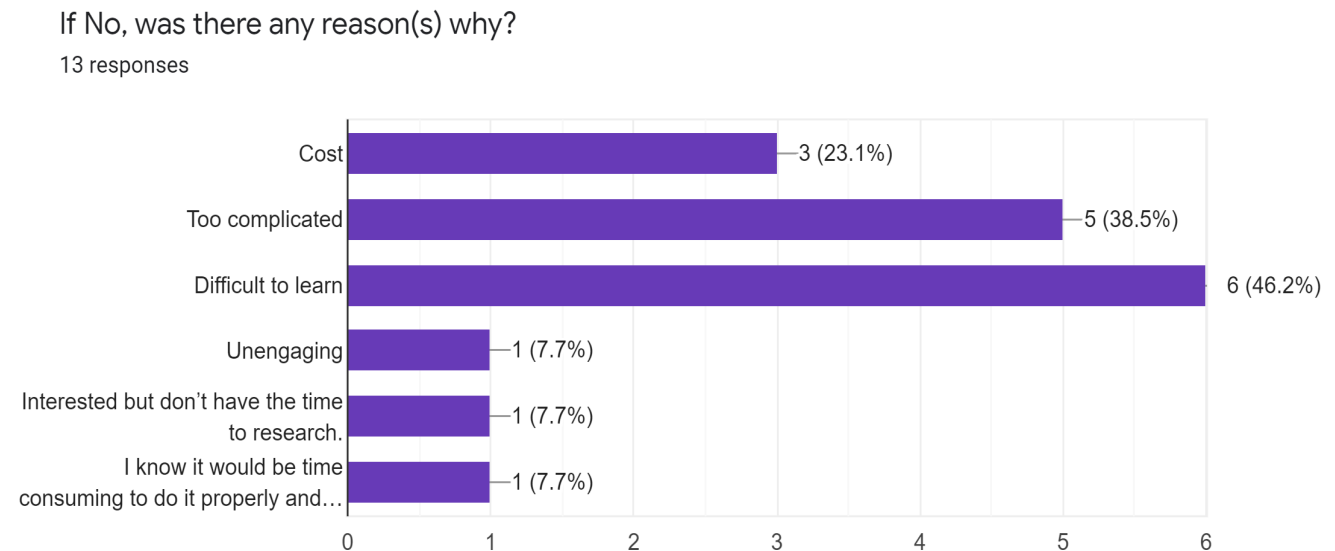
If Yes, was there any difficulties you faced?

8 responses



The survey responses to the question “If Yes, was there any difficulties you faced?”. The “Yes” in this case refers to if the user answered “Yes” to the question “Have you traded stock before?”.

Figure 10 - Survey: If no, reasons not



The survey responses to the question “If No, was there any reason(s) why?”. The “No” in this case refers to if the user answered “No” to the question “Have you traded stock before?”.

Figure 11 - Competitor Analysis Table

Competitor	Competitor Type	Market	Media	Price Model	Features and Offers
Robinhood	Indirect	United States	Use Social Media and adverts to publicise services.	Payment for Order Flow	0% Commission Fees Some gamified features
eToro	Direct	Global (excluding the United States)	Use Social Media and adverts to publicise services.	Fees (Spreads, Overnight, Weekend, Inactivity, Withdrawal and currency conversion)	0% Commission Fees

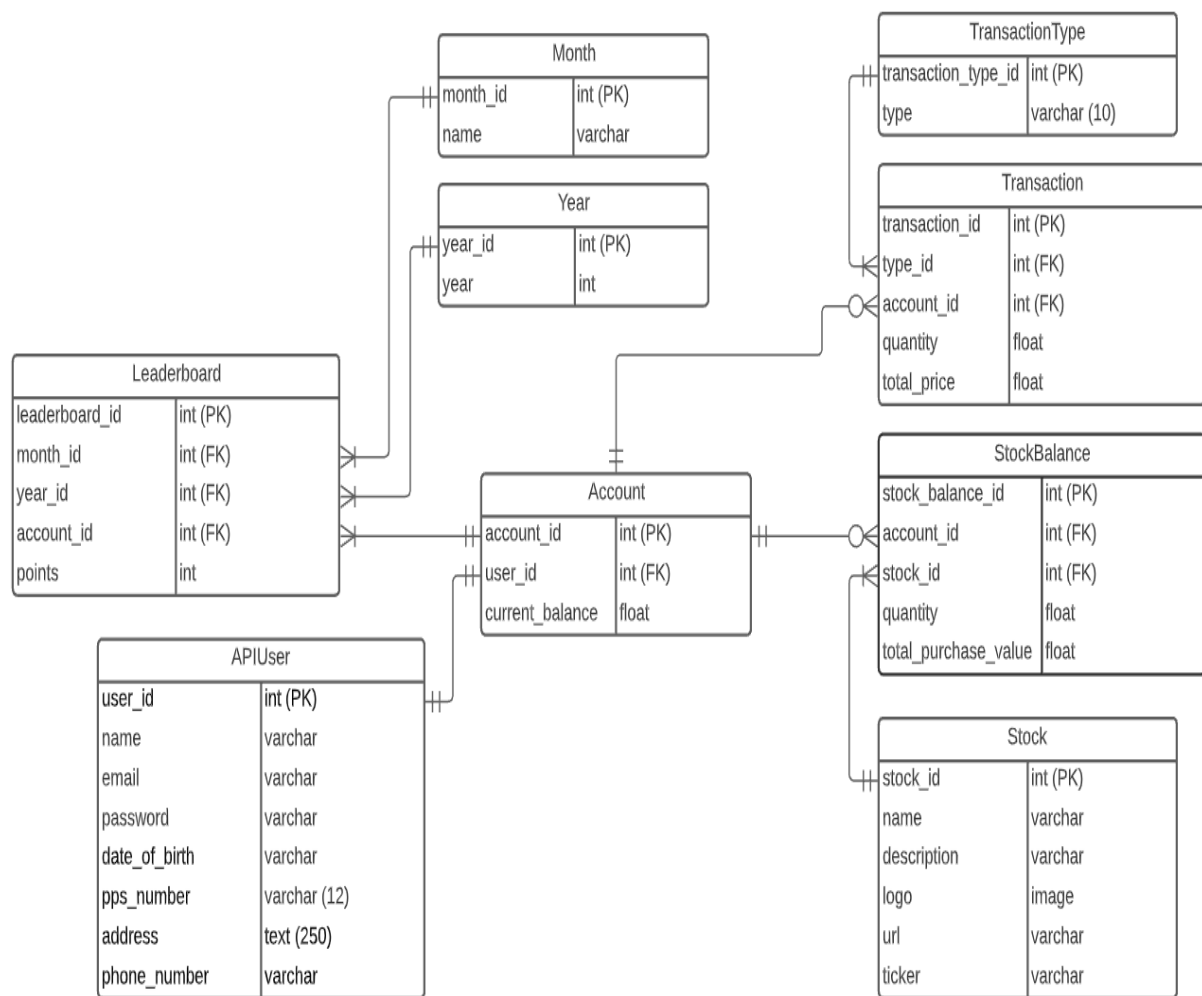
A table detailing two stock trading platforms we would be competing against.

Figure 12 - Financial Plan (First Year)

INVESTMENT INCOME	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEPT	OCT	NOV	DEC	TOTALS
Enterprise Ireland	€150,000	€0	€0	€0	€0	€0	€0	€0	€0	€0	€0	€0	€150,000
BOL loan	€0	€0	€0	€0	€0	€0	€0	€20,000	€0	€0	€0	€0	€20,000
TOTALS	€150,000	€0	€0	€0	€0	€0	€0	€20,000	€0	€0	€0	€0	€170,000
OPERATING REVENUE	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEPT	OCT	NOV	DEC	TOTALS
Commission Fees	€0	€0	€0	€0	€0	€0	€1,050	€2,100	€10,500	€21,000	€52,500	€105,000	€192,150
Subscriptions	€0	€0	€0	€0	€0	€0	€1,250	€2,500	€12,500	€25,000	€42,500	€125,000	€228,750
Ads	€0	€0	€0	€0	€0	€0	€8	€25	€50	€83	€166	€250	€582
TOTALS	€0	€0	€0	€0	€0	€0	€2,308	€4,625	€23,050	€46,083	€115,166	€230,250	€421,482
MONTHLY EXPENSES	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEPT	OCT	NOV	DEC	TOTALS
Salaries (Dev)	€13,500	€13,500	€13,500	€13,500	€13,500	€13,500	€13,500	€13,500	€13,500	€13,500	€13,500	€13,500	€162,000
Salaries (UI/UX)	€0	€0	€0	€0	€5,000	€5,000	€5,000	€5,000	€5,000	€5,000	€5,000	€5,000	€40,000
Salaries (Marketing)	€0	€0	€0	€0	€0	€5,200	€5,200	€5,200	€5,200	€5,200	€5,200	€5,200	€36,400
Heroku (Domain Hosting)	€50	€50	€50	€50	€50	€50	€50	€50	€50	€50	€50	€50	€600
Rent	€1,000	€1,000	€1,000	€1,000	€1,000	€1,000	€1,000	€1,000	€1,000	€1,000	€1,000	€1,000	€12,000
Equipment	€6,885	€0	€0	€0	€3,028	€2,110	€0	€0	€0	€0	€0	€0	€12,023
Misc	€500	€500	€500	€500	€500	€500	€500	€500	€500	€500	€500	€500	€6,000
User Prizes	€0	€0	€0	€0	€0	€0	€200	€200	€200	€200	€200	€200	€1,200
Loan Interest	€0	€0	€0	€0	€0	€0	€0	€0	€1,724	€1,724	€1,724	€1,724	€6,897
BURN RATE	€21,935	€15,050	€15,050	€15,050	€23,078	€27,360	€25,450	€25,450	€27,174	€27,174	€27,174	€27,174	€277,120
OVERVIEW	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEPT	OCT	NOV	DEC	TOTALS
CUMULATIVE EXPENSES	€21,935	€36,985	€52,035	€67,085	€90,163	€117,523	€142,973	€168,423	€195,597	€222,771	€249,946	€277,120	€277,120
CUMULATIVE INCOME	€150,000	€150,000	€150,000	€150,000	€150,000	€150,000	€152,308	€176,933	€199,983	€246,066	€361,232	€591,482	€591,482
STATUS	€128,065	€113,015	€97,965	€82,915	€59,837	€32,477	€9,335	€8,510	€4,386	€23,295	€111,286	€314,362	€314,362

A spreadsheet detailing the financial plan for the first year of the business.

Figure 13 - Database (ERD)



This Entity Relationship Diagram gives a planned representation of how our Django Models will be structured, the content they will hold and the links between them.