# INFS3202 Project Proposal
# Financial Freedom

**Written by:**
Minh Nguyen 43552065
Nikolai Chaourov 43577554

Minh Nguyen 43552065 Nikolai Chaourov 43577554

## Describe the proposal

'Financial Freedom' is a browser based web-application designed for individuals interested in a simple and easy way of managing their personal finances. It provides a variety of budgeting features, visual analytics, and different avenues of notifications that will assist and remind the user to keep track of their spending and savings.

The system allows users to create transactions which will be compared to the budgets and saving goals they have specified. It also gives users insight into how their own financial position compares to their past performance as well as others in the community.

The system provides visual analytics into users spending and saving habits, which can be organised and filtered based on a variety of criteria's such as categories, time period, and amount.

Furthermore, 'Financial Freedom', will give users the opportunity to opt in to a notification system which will send emails, and SMS messages in response to unsustainable or irregular spending.

## Use cases

**Use Case:** Add Transaction
**Purpose:** Add a transaction to a list of the users existing transactions.
**Interaction:** The user would click on the '+ transaction' button, opening the transaction modal. The user would then fill out the transactional details (category, short description, amount, timestamp). Submitting the form would send the data to the web-API which would store it in the database in the user's transactions.

**Use Case:** Remove Transaction
**Purpose:** Remove a transaction from the users list of existing transactions.
**Interaction:** Open the list of users existing transactions, and remove the desired transaction by clicking the corresponding remove icon.

**Use Case:** Edit Transaction
**Purpose:** Edit the details of an existing transaction
**Interaction:** Open the list of users existing transactions, click on the transaction row to edit. This will open a transaction modal with prefilled values for the row.

**Use Case:** Add budget
**Purpose:** Create savings and budget goals which will be used as benchmarks for the comparison and analytics features.
**Interaction:** Click the 'Budget' button. This will send you to the extended create budget form. Here you will be able to add any recurring income and expenses, as well as savings goals. Submitting this form will POST the data to the API, which will store it in the database with reference to the user's account.

**Use Case:** Edit budget
**Purpose:** Edit the existing saving goals, and budgets.
**Interaction:** Click the 'Budget' button. If a budget already exists, all values on the page will be pre-filled, and an 'Edit' button will exist. Clicking the 'Edit' button will present the budget form, with all values pre-filled.

**Use Case:** Analyse transactions
**Purpose:** Create a graphical representation of the transactions for analytical purposes
**Interaction:** Going to the 'Analytics' page will present the user with options of data they can view. Clicking an option, such as 'Transaction analysis' will send a request to the API for a user's transaction data in a format that can easily be consumed by a graphing library, such as D3.js. The browser will then render the transaction data in several graphical forms, allowing the user to easily reason about their spending habits.

**Use Case:** Opt in/out of notifications
**Purpose:** Enable or disable available avenues of notifications to notify the users of irregular spending habits.
**Interaction:** The settings page of the application will have checkboxes for SMS and Email notifications. Checking the boxes and saving settings will opt a user in for SMS and Email notifications of unsustainable or irregular transactions.

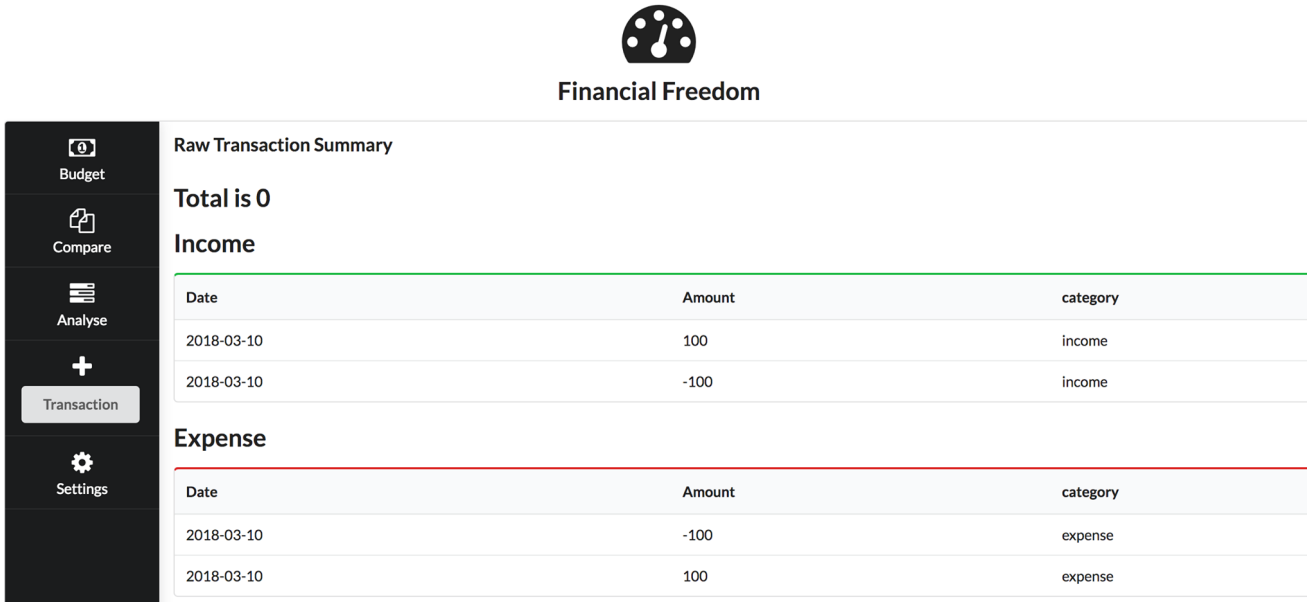
## Communication topology

The system will be implemented as a 3-tier architecture, consisting of a web-client, web-API, and relational database. The web-client will be written on the React framework (declarative client side framework), using MobX for application state management, and semantic-UI for styled components. The web-API will be a REST API, written in python with the help of the Falcon framework. Data will be stored in a PostgreSQL database, an open source relational database management system. The web-API will perform database operations using a combination of, the playhouse library's PostgreSQL adapter, in conjunction with the Peewee object relational mapper.

Minh Nguyen 43552065 Nikolai Chaourov 43577554

## Design Mock-up

### Transaction summary




### Create/Edit transaction modal