

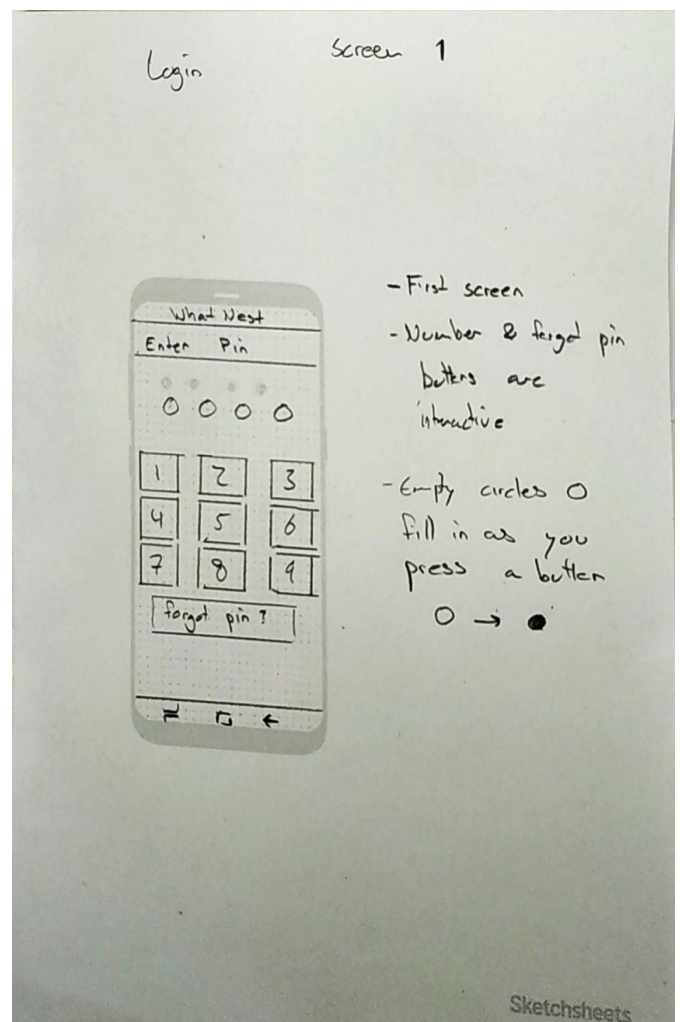
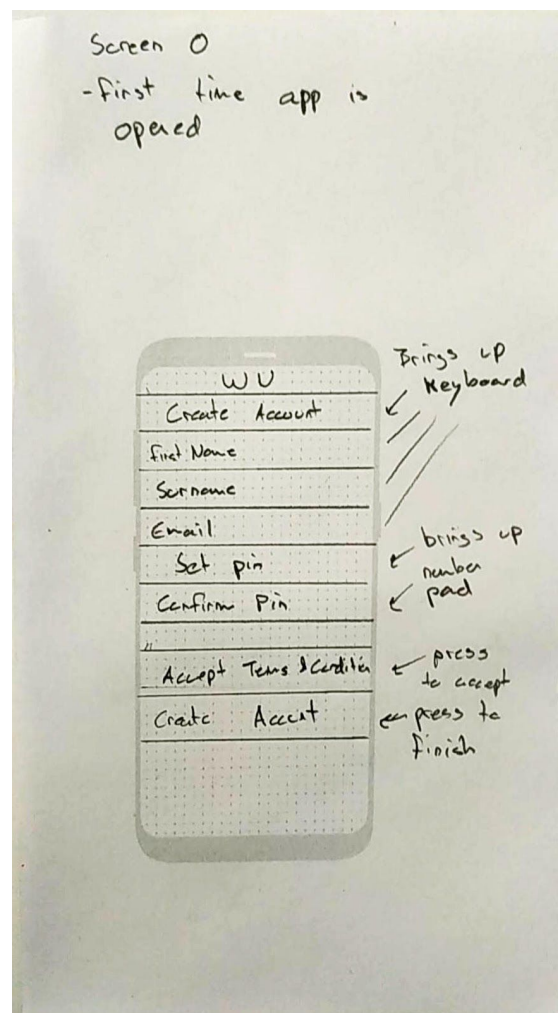
COMP1003 – Coursework 02 – Template

(You could extend your CW1 submission with this template – or upload a document based on this template only. Either is fine.)

Team Name: SegFault

1) Prototypes

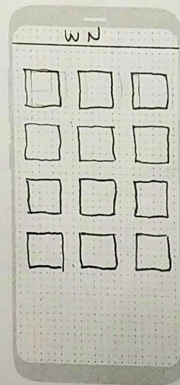
- convert any paper or tool-based prototypes into this document, via screenshots/photos



Galaxy S9

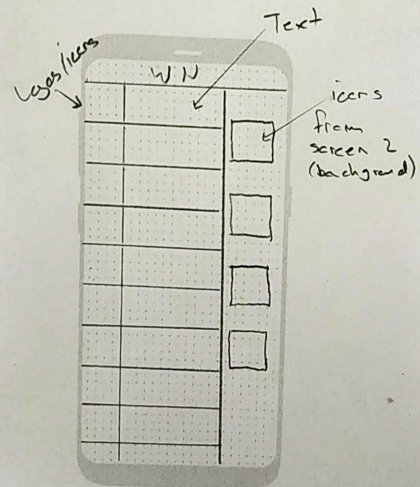
Screen 2 Home screen

- Home Screen
- Interactive buttons

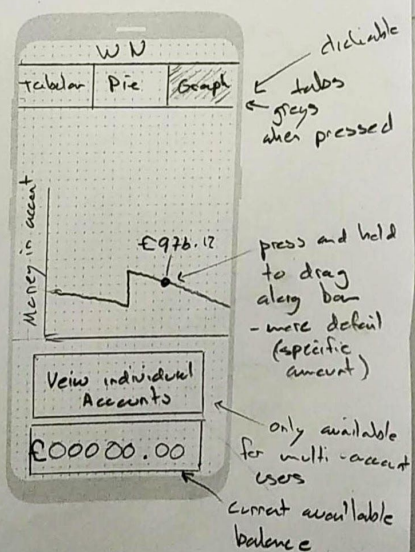


Screen 3

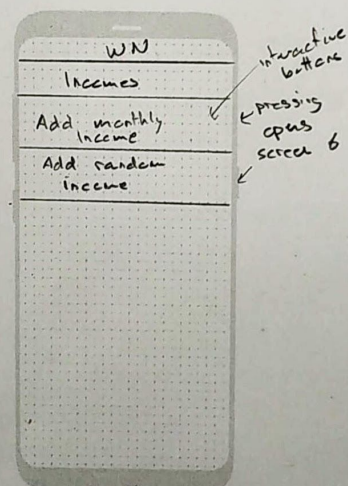
- Swipe right to open
- left hand side symbols match icons from screen 2
- more info about menus

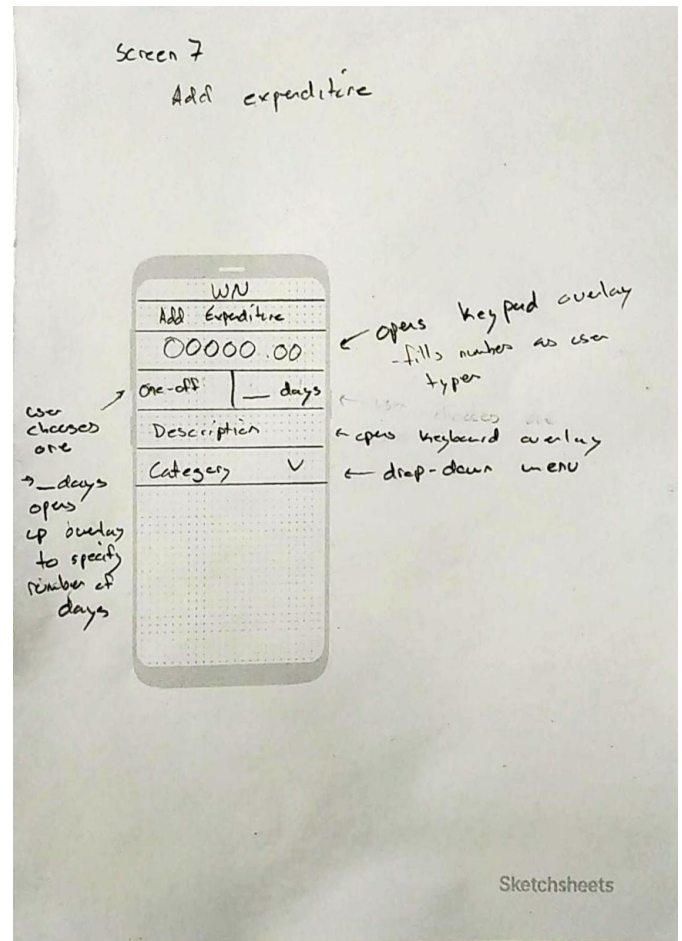
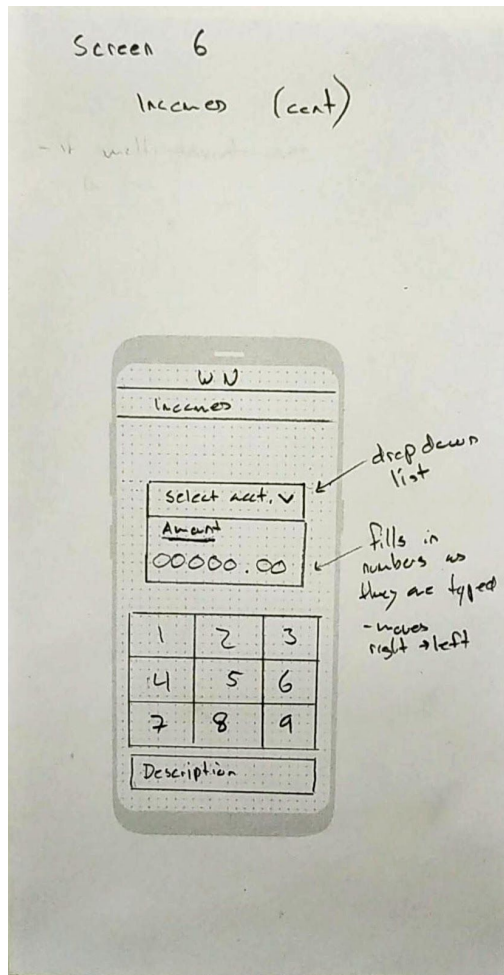


Screen 4 Expenditures



Screen 5 Incomes





Notes/Explanations/Unanswered Questions/Most Important Details:

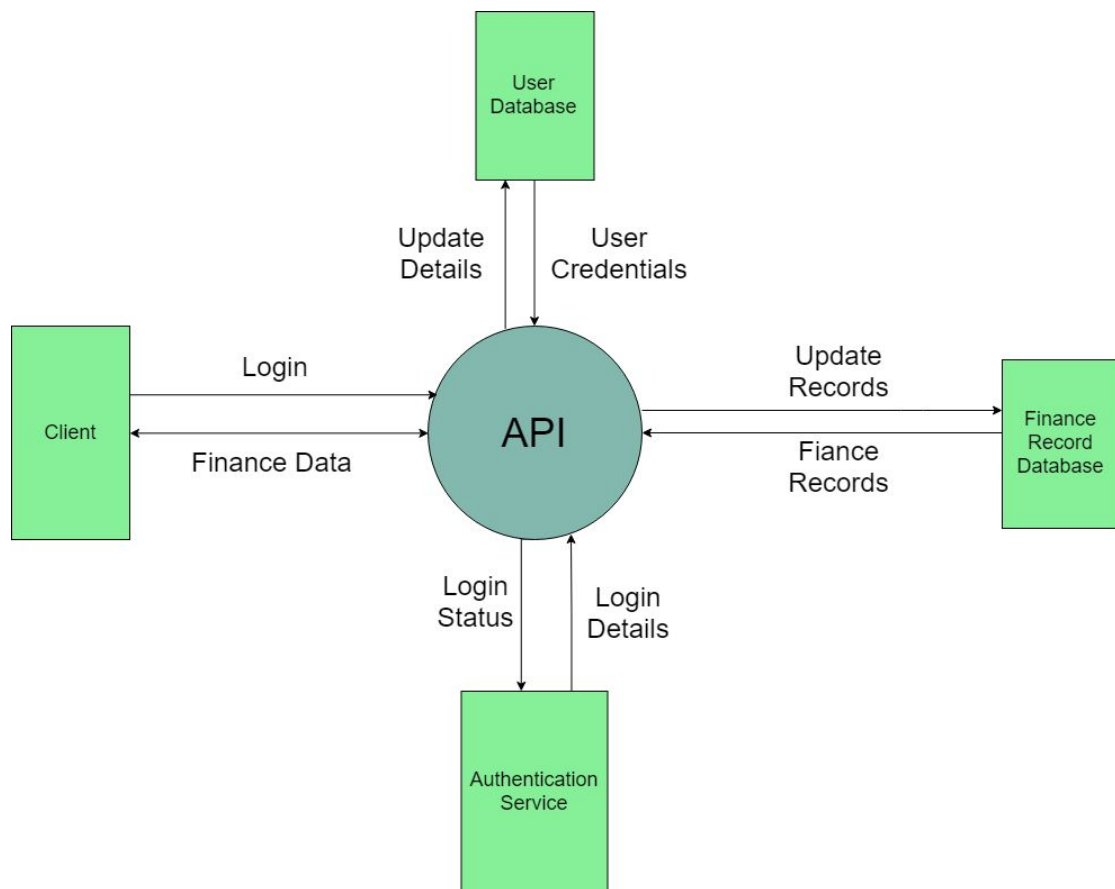
See notes on diagram pages.

2) Diagram Chosen: Context diagram

Reason for choice:

In order to simplify each individual process we decided to split up our proposed system into multiple sections, this makes each one easier to implement. In order to clearly draw separations between services we decided to use a context diagram to show that separation.

Diagram(s):



Notes/Explanations/Unanswered Questions/Most Important Details:

Notes:

- Diagram is very simple and many services could be split up into smaller services
- User should still be able to update records. Change expense category (food, clothing etc.) but isn't able to make transactions through the app.

Explanations:

- User makes login and data requests to the API and the API handles all the background tasks by talking to the internal services
- For login the API requests login details (status; is the user logged in already?) from the authentication services and is also able to update it (user logs out/in). If the user tries to log in it firsts check the user database and then checks if the details are correct.
- API can also change the user details (change password/username).

Questions:

- Should the user be able to upload receipts and possibly have the system recognize the store and expense category? Or should we have a store database and track all transactions to retailers?

3) Diagram Chosen: Scenarios

Reason for choice:

There are many parts to this system and many different things that a user can do. We create scenarios to plan what is the simplest method to do things for a user, allowing for us to build our aforementioned prototypes.

Diagram(s):

Scenario:1	
Title	Get overview of expenditures
Overview	The user will be able to see, both graphically and in numerical format, a breakdown of what they have spent money on.
Rationale	This is done to fulfil one of the use cases that we found during requirements analysis - that they must be able to get an overview of their expenses. This means that the user will be able to see, separated by spending category what they have spent money on.
Detail	<ol style="list-style-type: none"> 1. User opens app 2. User logs in by entering their pin 3. User selects the "Expenditure Overview" button 4. Graphs showing user's expenditures appear on screen
Notes	<p>There will be multiple ways of accessing the expenditure overview. You could either get it from the sliding menu or from the main menu.</p> <p>Pin is not their bank account pin, it is a pin to log in to the app for security purposes.</p>

Scenario: 2	
Title	Add monthly income
Overview	User will be able to enter their monthly income in to the app
Rationale	This is done so that the user can record how much money they earn each month. They will then be able to use this to manage their budget or compare the income of different months
Detail	<ol style="list-style-type: none"> 1. User selects "Update income" button 2. User is able to enter monthly income in numerical form 3. User enters date that money was/will be paid in to account
Notes	Assuming user starts from home page

Scenario: 3	
Title	Get single summary of all bank accounts
Overview	The user will be able to view collective information of all their bank accounts together. This includes total funds available, total spending and more. The user is also able to get individual information for each of their accounts as well.
Rationale	This is done so that the user can view their bank informations with ease without any calculation. The user is also provided with easy access to all bank accounts at once so that they have a master access to each of them individually to view or manage.
Detail	<ol style="list-style-type: none"> 1. User opens app 2. User logs in to their account 3. User selects "Expenditure overview" option 4. User is asked for additional safety questions 5. Information, calculations and graphs are presented to user across all accounts with a choice to view individual accounts 6. User presses "View Individually" option and chooses one or more accounts to view separately. 7. User can simply swipe left to view opened individual accounts one by one and scroll down for detailed information on each.
Notes	Assumes that the user is a Multi-Account user in order to access this option. Otherwise, this option is not available.

Scenario: 4	
Title	Add random income
Overview	In addition to user entering monthly income, user is also able to enter extra random income. E.g. a gift from a friend
Rationale	This is done so the user can see how much money they currently have - not just the regular incoming amount, which fulfils one of our user requirements
Detail	<ol style="list-style-type: none"> 1. User clicks on "Enter extra income" button 2. User selects which account to pay in to 3. User enters extra income in numerical form 4. User can enter a short description 5. User can enter the date the money was/will be paid in to account
Notes	Assuming user starts from home page

Scenario: 5	
Title	Add an expenditure
Overview	The user will have spent money, and will want to add it to the system so that it has been tracked.
Rationale	In order for the user to keep the system up to date and provide expenditure, they must be able to
Detail	<ol style="list-style-type: none"> 1. User clicks on “Add expenditure” button 2. User can enter amount that has been spent 3. User needs to enter type of expenditure using a drop down menu 4. Amount is deducted from available income 5. Expenditure has been added to the list of expenditures
Notes	<p>Assumes user starts on home screen.</p> <p>If type is not included in drop down list, ‘other’ can be selected and user can enter their own type</p>

Scenario: 6	
Title	Add another bank account or joint-account
Overview	The user can add more accounts to the application for easier viewing and managing in the future.
Rationale	This is done so that Multi-Account viewing is possible as well as being able to specify joint-accounts.
Detail	<ol style="list-style-type: none"> 1. User opens the home page 2. User selects "Add Another Account" option 3. User specifies if the account is shared or not 4. User inputs desired name and the account is created at a default of zero funds and spending. 5. User can proceed to manage their accounts as normal.
Notes	The amount of detail the account holds is upto the user. They can always add additional details as they please where each account is unique and their settings to do not overwrite.

Scenario: 7	
Title	Set aside amount for 'spare money'
Overview	User can choose to keep an amount of money separate from the 'available money' section.
Rationale	This allows the user to view how much extra money they have to spend. This is useful as it allows user to have a safety net in case a large bill suddenly needs paying or an urgent repair needs to be made. This also fulfils one of our requirements
Detail	<ol style="list-style-type: none"> 1. User clicks on "Set aside money" button 2. User enters amount to set aside in numerical form 3. Amount is deducted from available amount to spend
Notes	Assume user starts on home page

Scenario: 8	
Title	Create initial account
Overview	User can create an account so that details can be entered about the account in order to start saving/viewing income/outcome.
Rationale	When first opening the app, the user will need to enter details about an account in order to start viewing/updating expenditure etc.
Detail	<ol style="list-style-type: none"> 1. User clicks on "Create Account" 2. User enters account details 3. User presses submit button
Notes	View the specific details that a user can enter by looking at Screen 0 in our paper prototypes

Scenario: 9	
Title	Regular expenditure
Overview	User can set custom categories and sums in which their “Spending Money” or “Savings” are automatically deducted from
Rationale	This is used so that users can easily count food, rent and similar budgeting costs automatically without having to deduct each and every one of them manually all the time.
Detail	<ol style="list-style-type: none"> 1. User chooses the “Create Regular Expenditure” option 2. User sets the name of their expenditure and can add a description. 3. User sets the amount of money that is being spent 4. User sets certain amount of days and as a result, the app automatically deducts the expenditure set per the days set both inputted by the user
Notes	User can set a certain amount of days of repetition, but not a certain day of the month at the moment.

Notes/Explanations/Unanswered Questions/Most Important Details:

Any decisions on design we made in the scenarios are in the notes sections of each scenario. Please consult them for our notes and most important details.