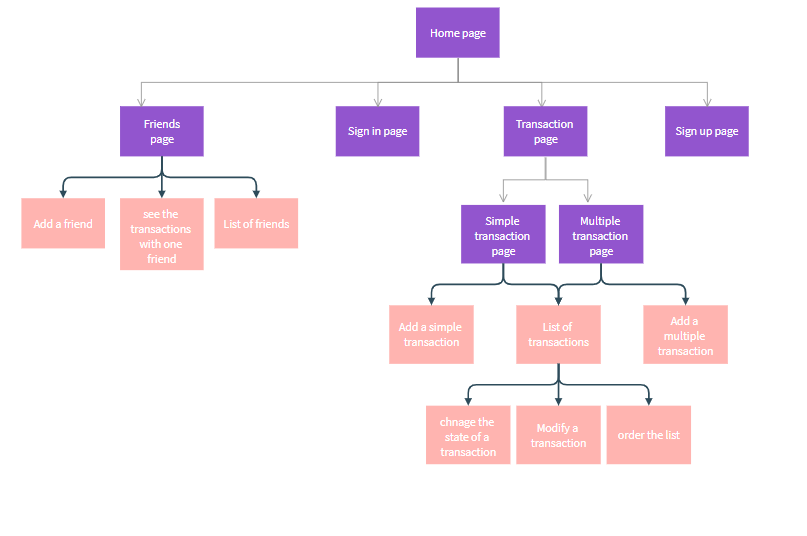
|  |  |
| --- | --- |
|  | **Programmation Web**  ⌘ Documentation template |

**1. Target user profiles**

*Persona profile.*

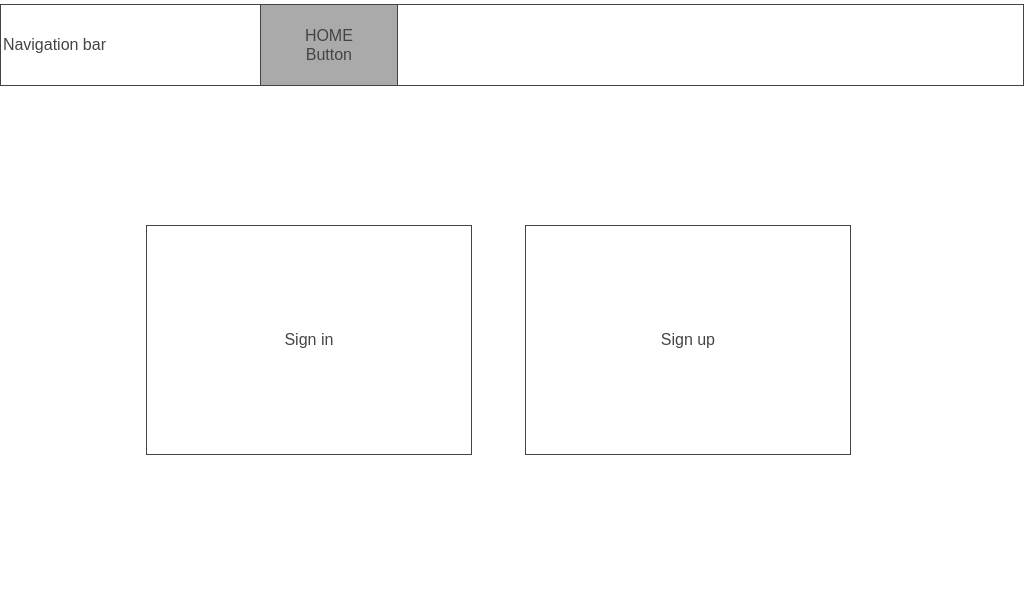
* *Name: David*
* *Age: 23*
* *Commonly used applications: Lydia, Tricount.*
* *Back story: Being a student he often goes to the bar with his friend, where he or his friend often pays rounds of beer, but he like to keep track of who paid what to whom.*
* *Overview:*   
  *- loves maths, numbers & precision,*  
  *- like to know how much money he spent and lended*

**2. Sitemap**

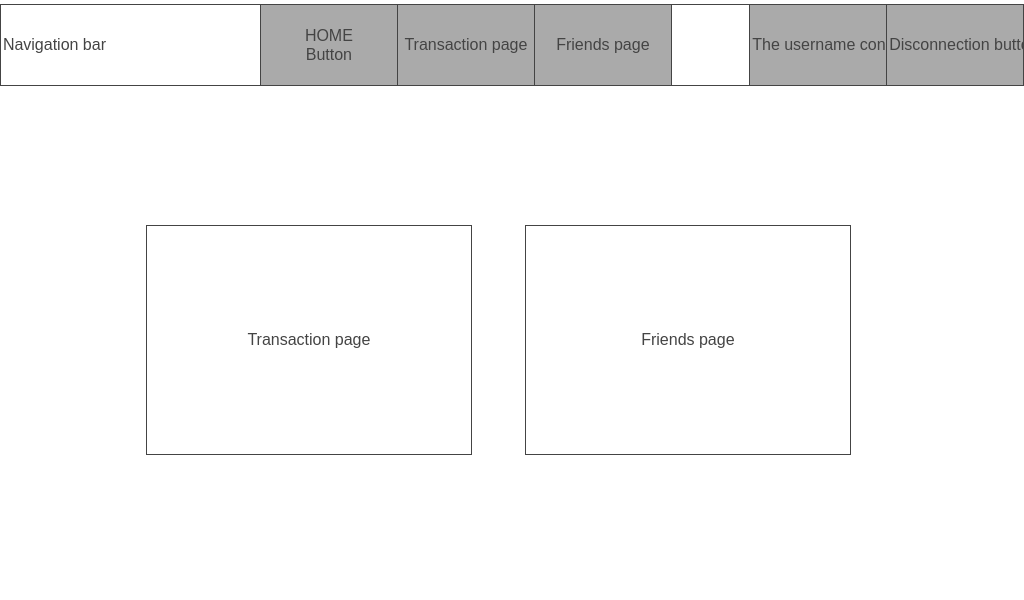


**3. Wireframes**

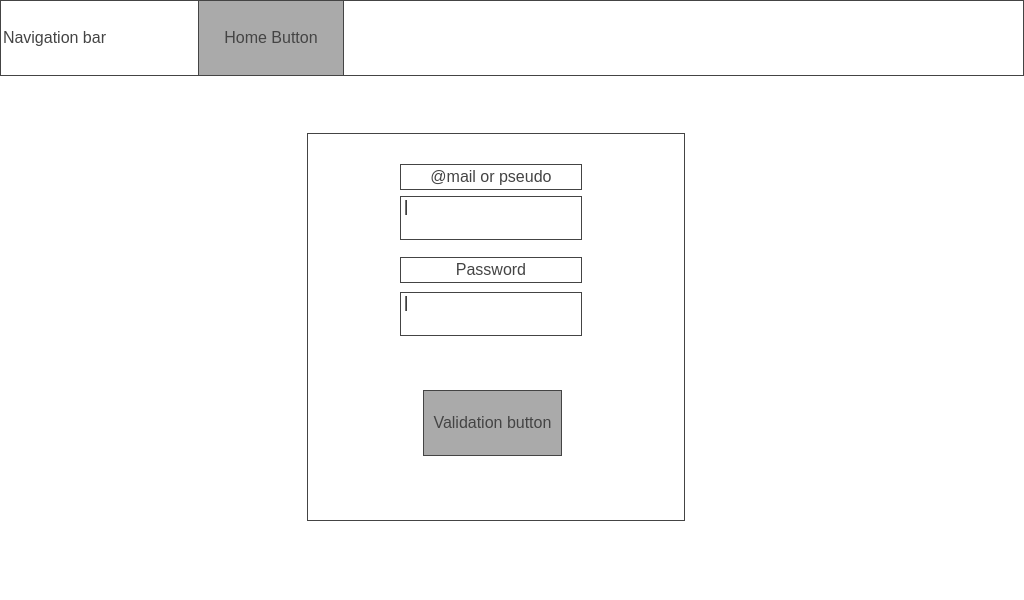
**Home page when the user is disconnected**



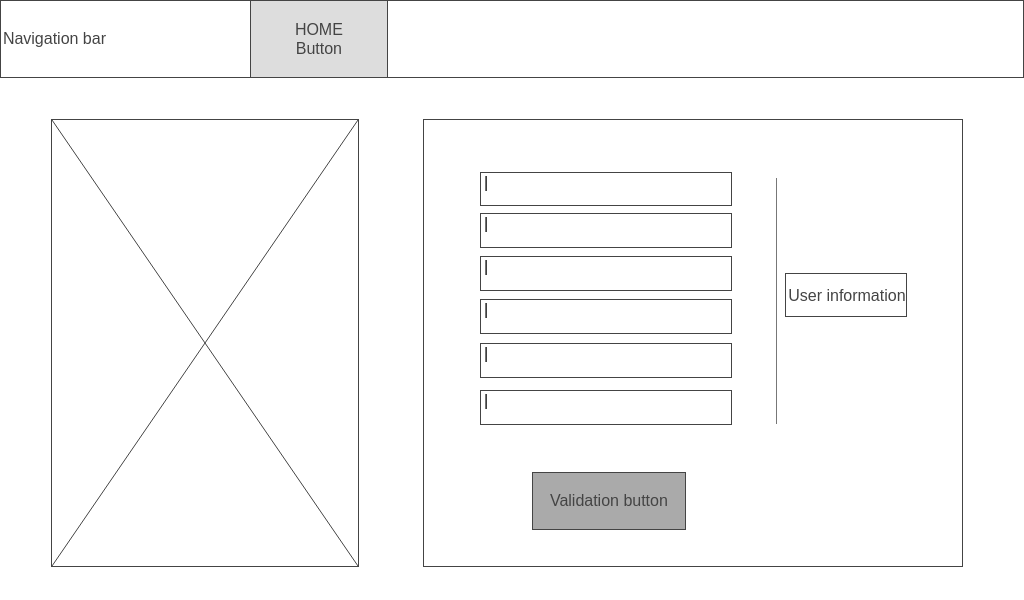
**Home page when the user is connected**



**Sign in page**



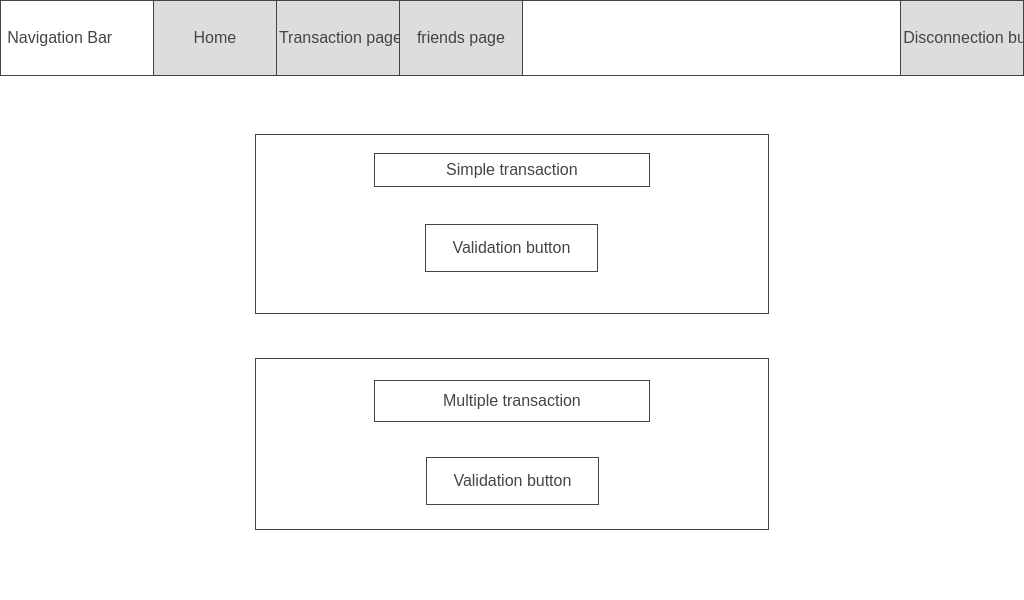
**Sign up page**



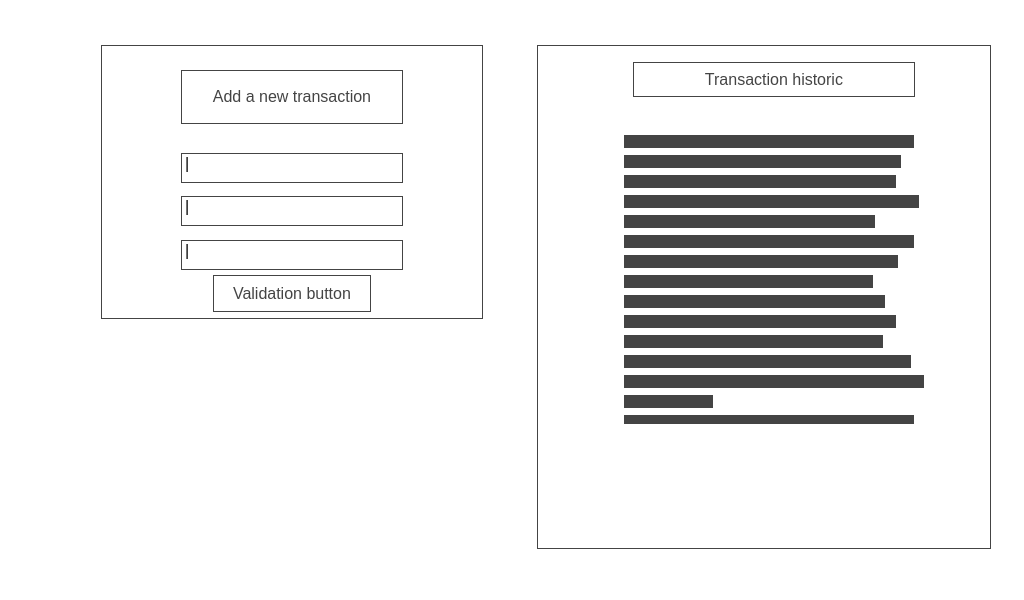
**Friends page**



**Transaction choice page**



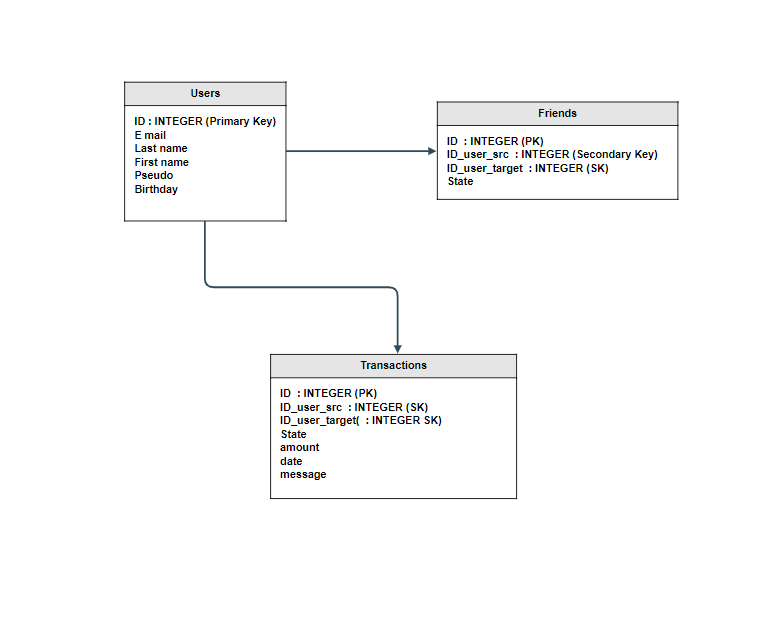
**Simple transaction page**



**Multiple transaction page**



**4. Database model**



**5. System structure - scripts overview**

Page Home

- index.php : Wireframe home page

- Page d’accueil du site

-Méthode Get pour afficher les deux versions possible: version connectée et deconnectée

Connexion

- connexion.php, check\_login.php : Wireframe sign in page

- Page de connection à l’application, formulaire simple.

- Méthode POST pour verifier les données de l’utilisateur dans la table Utilisateurs vers check\_login.php

-Méthode Get pour afficher les messages d’erreur d’identification

Inscription

- page\_creation.php, validation\_creation.php : Wireframe sign up page

- Page d’inscription à l’application, formulaire demandant toutes les informations

- Méthode POST pour verifier et valider l’implémentation des données du nouvel utilisateur dans la table Utilisateurs vers validation\_creation.php

-Méthode Get pour afficher les messages d’erreurs de formulaire.

Ajouter un ami:

- ajouter\_ami.php, validation\_ajout\_ami.php : Wireframe friends page

- Page d’ajout d’ami et d’affichage des demandes d’ami/liste d’amis

- Méthode POST pour verifier et valider l’implémentation des données de la demande d’ami dans la table Amis vers validation\_ajout\_ami.php

-Méthode Get pour afficher les messages d’erreurs de formulaire.

Accepter/Refuser une demande d’ ami:

- ajouter\_ami.php, action\_ami.php: Wireframe friends page

- Page d’ajout d’ami et d’affichage des demandes d’ami/liste d’amis

- Parcours de la Table Amis et affichage des demandes d’ami non traitées.

-Méthode get pour ajouter ou supprimer une demande d’ami via action\_ami.php

Afficher les transactions avec un ami:

- ajouter\_ami.php, transactions\_ami.php: Wireframe friends page

- Page d’ajout d’ami et d’affichage des demandes d’ami/liste d’amis

-Méthode get pour voir les transactions avec un unique ami via transactions\_ami.php

Page d’accueil des transactions:

- gestion\_transaction.php, : Wireframe Transactions choice page

- Page de choix entre transaction simple ou mutliple

-Dirige vers la page transactions.php pour simple et transactions\_grp.php pour multiple

Ajout transactions simples:

- transactions.php, validation\_ajout\_transaction.php : Wireframe simple transactions page

- Ajout d’une transaction simple et affichage de l’historique des transactions

-Méthode Post pour verifier les données du formulaire via validation\_ajout\_transaction.php

-Méthode Get pour afficher les messages d’erreurs.

Modification d’une transaction:

- transactions.php, action\_transaction.php : Wireframe simple transactions page

- Ajout d’une transaction simple et affichage de l’historique des transactions

-Méthode Get pour choisir quelle partie on veut modifier via action\_transaction.php

-Méthode Post pour rentrer la modification via action\_transaction.php

Ajout transactions multiples:

- transactions\_grp.php, validation\_ajout\_transaction\_grp.php, pre\_validation\_ajout\_transaction\_grp.php : Wireframe multiple transactions page

- Ajout d’une transaction multiple et affichage de l’historique des transactions

-Méthode Post pour verifier les données du 1er formulaire concernant le nombre d’utilisateurs cibles et la méthode de remplissage du montant choisie via pre\_validation\_ajout\_transaction.php

-Méthode Post pour verifier les données du 2eme formulaire concernant la validation des données choisies via validation\_ajout\_transaction.php

-Méthode Get pour afficher les messages d’erreurs et le mode automatique ou manuel choisi.

Création de la BDD et des tables:

- install.php

**6. Test strategy**

Test strategy presents the approach to testing taken on a given project. It describes the scope of the tests, the methods used and specification of test cases.

**6A. Test items**

This section presents a list of elements of the system that you intend to test within the scope of the test plan. Essentially, it is a list of what is to be tested - application components.

* Authentication Graphical User Interface
* Friends booklet
* ….

**6B. Test cases**

There exist different types oftests - functional, integration, etc. - you are free to formalize the test cases that you find appropriate to ensure the right level of quality of the project.

The design of the **functional testing** will focus on the adherence of the system to **user requirements**. Integration testing assures that the associated modules communicate correctly, hence tests of inputs and outputs of the system need to be carried out to verify that they conform to the system specifications.

Each test is executed and the result is either a pass or fail.

Test case ID: #1

Test case Name: User Login

Test case Date:

Test case Purpose: To ensure that only valid users are able to log into the application

Test case Function: To verify that only a correct/valid combination of username and password allow a user to log into the application.

Test case Input: An alphanumeric username and an alphanumeric password

Test case expected Output: For a registered user entering the correct combination of username and password, the control panel of the application is shown. For an incorrect combination, the user remains on the login page and an appropriate error message is shown.

**6C. Requirements Traceability Matrix (RTM)**

RTM documents the links between system requirements and test cases that are put in place to ensure an adequate level of testing is achieved. The main purpose of Requirement Traceability Matrix is to see that all requirements have their corresponding test cases so that no functionality is missed out while performing software testing.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REQUIREMENTS TRACEABILITY MATRIX** | | | | | |
| **Functional Requirement ID#** | **Functional Requirement / Use Case** | **Test Case ID#** | **Test execution date** | **Execution Status** | **Defect description** |
| FR\_1 | As a system user I can add a friend to my friends booklet | TC#001  TC#002 | 20/03/2020 | pass |  |
| FR\_2 | As a system user I can modify existing transactions with a friend | TC#003  TC#004 | 21/03/2020 | fail | The transaction date field can be modified |
|  |  |  |  |  |  |