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

**Instructions**

Please fill in all the sections of the document. Every section includes guidelines on how to fill it in and some of them contain samples and examples that are marked in blue. The document can be written in French.

**1. Target user profiles**

Describe the intended user base of the proposed system by defining one or more user personas. A **persona** is a fictional character who uses the product in a similar way to a potential user type. Think of the attributes of the user community.

*You can use the following sample persona profile template and enrich it with information you desire.*

*Persona profile.*

* *Name: John*
* *Age: 22*
* *Commonly used applications: Matlab, Tinder.*
* *Back story: a high-achiever and a Cambridge graduate that specializes in evolutionary algorithms, John moved to Bordeaux for an exciting job opportunity despite barely knowing the local language. John likes to go out and meet people but given his forgetful nature often in a rush leaves his wallet/credit card at home and ends up being unable to pay for his drinks.*
* *Overview:*   
  *- loves maths, numbers & precision,*  
  *- appreciates intuitive design and innovative solutions,*  
  *- is looking for an app that will help him keep track of expenses shared with other people.*

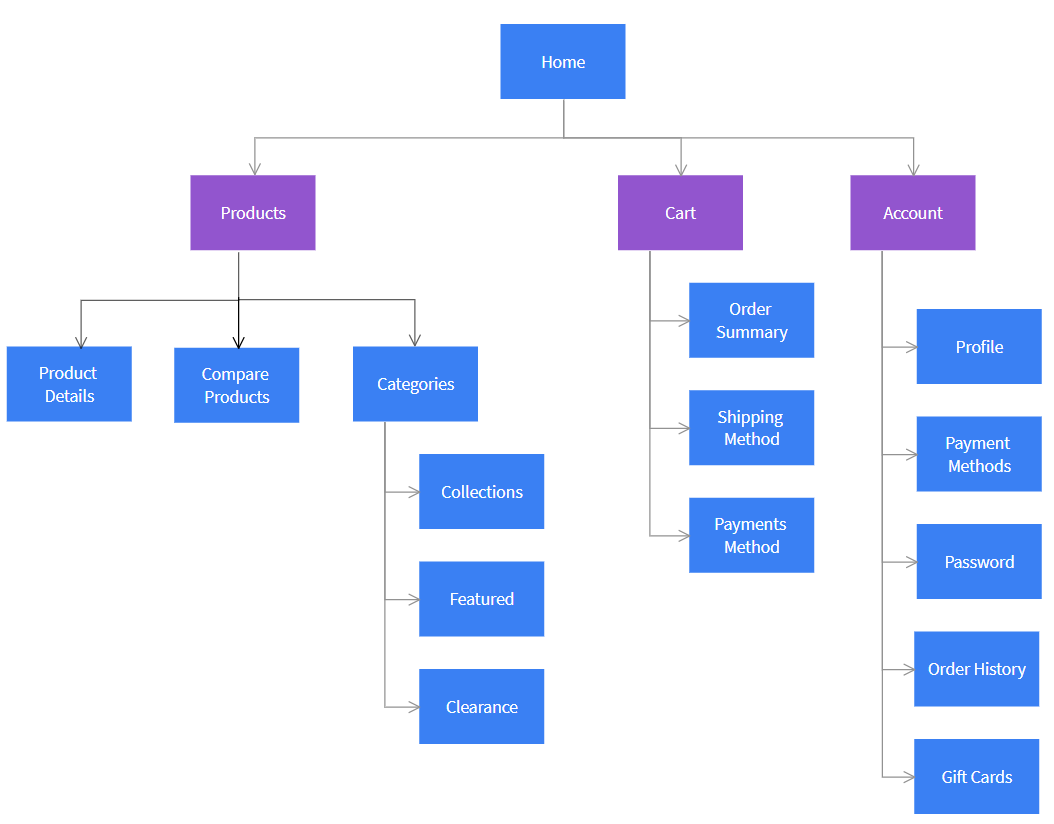
**2. Sitemap**

A site map is a diagram of a website’s pages, organized hierarchically. It makes it easy to visualize the basic structure and navigation of the website.

Showing key cross-links can be useful on a sitemap. A cross link is when one page links to another, but the pages don’t sit within the same navigation category.

Show if there are any changes to pages if the user is logged in. Do the users see a different version of the home page? Do they go to an entirely different section of the site all together? It’s important to visually separate any logged in areas of the sitemap so the developers can understand which users get access to which pages.

If you decide to use different key templates for subpages/elements of the target application, provide the project team with a checklist of all the key templates that need to be produced. Your sitemap needs to show which pages share the same template, e.g. home page might be one template; there might be several second-level category pages that all share the same template. Show this by placing a template number on each page in the sitemap. Pages that share the same template should have the same template number.



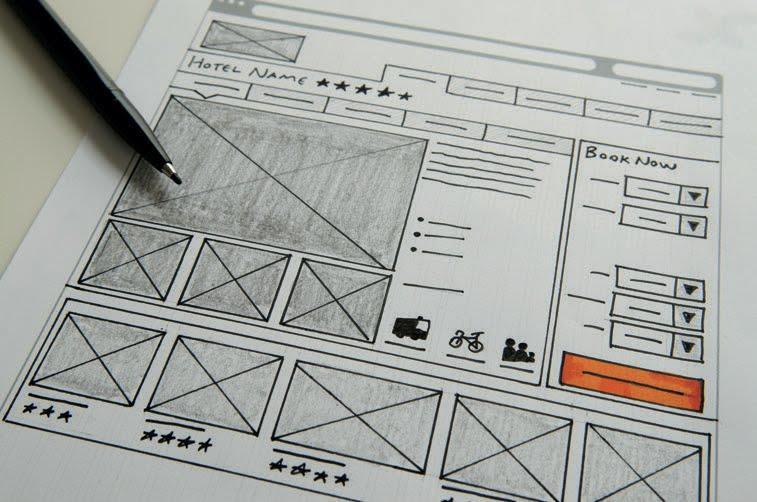
*A sample visual sitemap of an e-commerce website generated with moqups.com.  
It uses boxes and lines to show connections between pages, the placement of boxes shows the structure.*

**3. Wireframes**

A wireframe shows the UI elements (text, images, buttons, links etc…) that make up a screen, page or UI component. A well-presented wireframe helps the project team understand the ideal design and functional solution.

Developers need to know the content organization, responsive interactions (how something behaves or feels when you interact with it) and how user can manipulate the data before they start implementation. Wireframes respond to that need and provide structure for every page, key elements and their interactions as well as hierarchy of items.

Wireframes can be simply sketched using a pencil or a dedicated tool, go [here](https://uxplanet.org/5-free-quick-wireframe-tools-for-ui-ux-designers-in-2017-189e6a594fda) to see some suggestions.



*A sample wireframe sketched with a pencil [source: R. Caddick, S. Cable: Communicating the user experience - a practical guide for creating useful UX documentation]*

**Design recommendations for wireframes\***

The following elements should be considered when creating wireframes:

* **Layout**: You can organize the structure of the page and where the different elements sit. Content and images: What are the content and image requirements for each page? Where will you source the assets from?
* **Priority**: Wireframes help identify what the priority or key element of each page should be. Where do you want users to look? What do you want them to do?
* **Navigation**: How is a user going to move around the site? Wireframes let you try and test different approaches for navigation.

For every wireframe, do the page structure, navigation and workflows here:

* present information users want and expect?
* present information in a way where they can easily find what they need?
* use labels users will readily understand?
* allow users to easily and accurately predict the outcomes of their actions?

**Validating wireframes**

Get your wireframes in front of people informally as soon as possible. Friends, family, and colleagues will all help to rethink and improve your work. It’s never too early to test. Testing a simple sketch on a piece of paper can transform the success of your product.

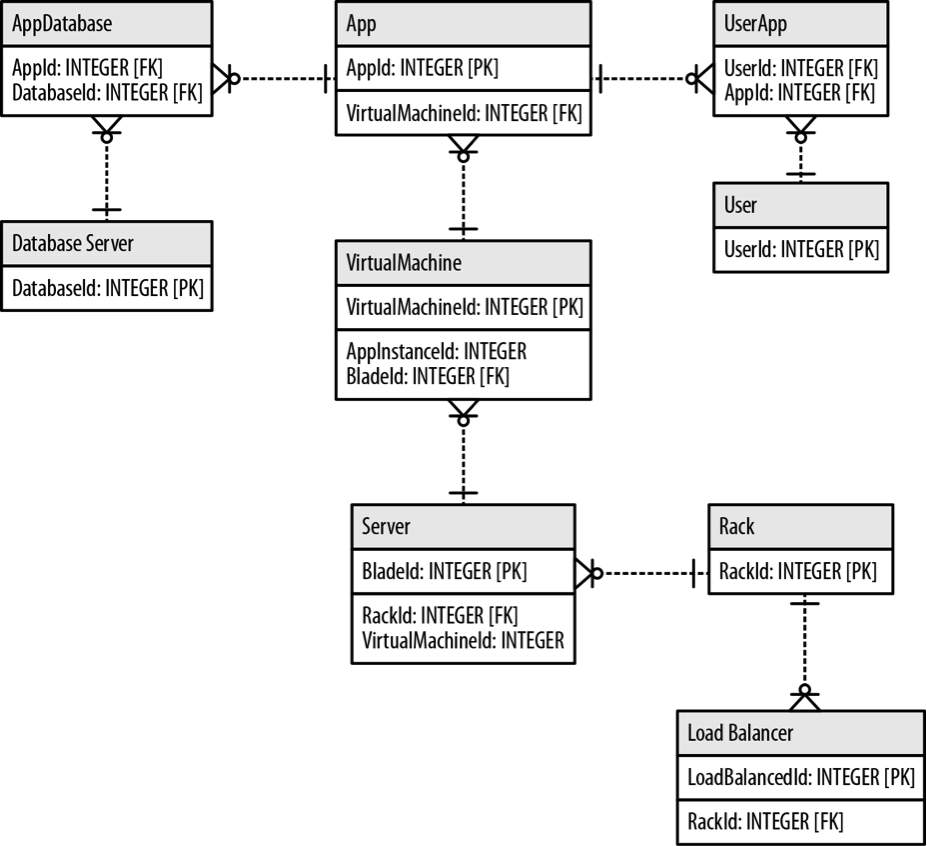
\*source: R. Caddick, S. Cable: Communicating the user experience - a practical guide for creating useful UX documentation

**4. Database model**

A database model determines the logical structure of a database and describes in which manner data can be stored, organized and manipulated. The most popular example of a database model is the relational model, which uses a table-based format.

**Obligatory elements of a database model:**

* Name and structure of the table, listing its fields and types
* Primary, secondary and foreign keys marked
* Relations between the tables and their cardinality

  
*A sample database model representing all the elements listed. Different, standard notation can be used to depict cardinality and table keys.*

**5. System structure - scripts overview**

For every php script created, please specify the following elements:

* php script name and corresponding Wireframe, if applicable,
* a couple of sentences describing the functionality of a given script ,
* HTTP method call and its parameters,
* type of interaction with the database (Insert/Delete/Update), if applicable,
* dependencies with other scripts, if applicable.

Connexion

- sign\_in.php, wireframe 1

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

- Méthode POST avec paramètres email, password vers connexion.php

Inscription

- sign\_up.php, wireframe 2

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

- Méthode POST avec paramètres email, username, firstname, lastname, birthdate, password

vers register.php

- Insert à la table “user”

…...**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 |
|  |  |  |  |  |  |