System Requirements and Architecture

1. System Requirements

According to our research and preparation, we have found that to achieve all our goals we will need to have proficiency in java; javascript and html. To be more specific we will be using our java skills to develop a secure password generating algorithm, the backbone of our project, and an android mobile application; javascript and html will be used towards creating a chromium web extension.

1.1. Requirements Elicitation

The first thing we did in our project was to define its main goals and decide it's scope. After discussing it as a group we decided to create at least a chromium browser extension and an android app, with the possibility of also developing a firefox extension and a Windows App depending on time constraints.

To achieve our goals we looked at projects with similar technologies, in our case, the project orienter had already guided us to a great application called "*LessPass".* Similarly to our project, *LessPass* computes a unique password using a site, login, and a master password, it also doesn't need to sync a password vault across every device or to the cloud, because it works offline. We are hoping to replicate what *LessPass* does but with even better security and an option to sync usernames across devices. In the end this research proved very worthwhile as what we found is open-source software that allowed us to learn a lot about what we need to do.

In order to decide what features we should implement (that weren't already obvious) we met with both our project orienters, who happen to have a background in cybersecurity. In that meeting we got a lot of suggestions on how to develop the basic modules of our project and where we should implement them, for example, it was suggested that we focus on developing a chromium web extension as it is far easier than a firefox one. The orienters also advised us to use our own personal experience with password managers to implement features that we think would improve the general experience.

1.2. Context Description

The way our service works the user will have to enter a master password, their username and the website's url in order to get his password, seeing as our algorithm generates replicable passwords at no point will they have to save it. We also have a

feature to change the received password by using a counter number, if the user changed the password “x” times they will have to change the counter to “x”.

Ideally we will have developed both a web extension and a mobile app by our final deadline, that means that any user that wishes to use complex and secure passwords will be able to get their credentials anywhere he goes. A person could create an account on a store’s website at their own home, travel to said store in person and get their login information even if they don’t have internet on their phone. That's just one of the many situations that have happened to us and where a service like the one we’re creating would have been extremely useful.

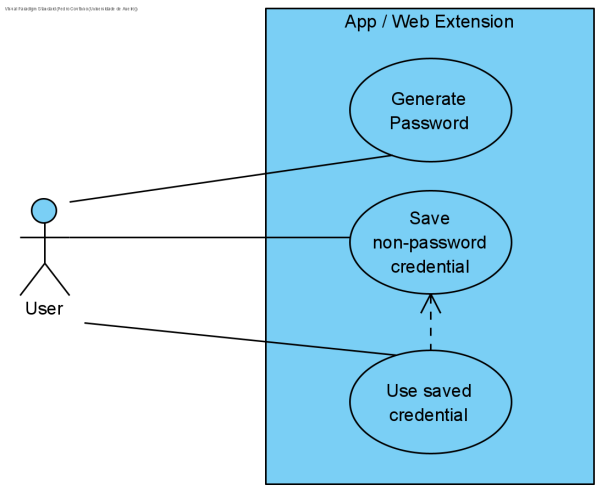
1.3. Actors

The target user for the browser extension is anyone that has the necessity to have various safe logins/passwords for multiple websites.
The user won’t need to have a high level of expertise, in fact, quite the opposite as a very basic understanding of their personal browser should be enough to use the extension.
The only actor for the extension is the following:

- **User:** represents the user of the extension. Will have access to the main feature which allows them to generate a safe password based on the website they’re logging in, username and a master password.

1.4. Use Cases

The following diagram presents the use case model of the system, we opted to do only one use case for both the mobile app and the web extension as they behave identically.



1.5. Non-functional Requirements

- **Security:** This is the whole purpose of the extension. The first thing anyone using the extension will expect is security. **Priority:** Very High.
- **Reliability:** The need for the service to be always available is very straightforward. Not having the credentials available at any time can be a huge problem for the user. **Priority:** High.
- **Usability:** Since the user will not want to waste any time logging in to websites, it is required that the service will be as simple as possible for fast use. **Priority:** Medium.
- **Portability:** Since there are various different browsers, it would be desirable that the extension is available for multiple browsers

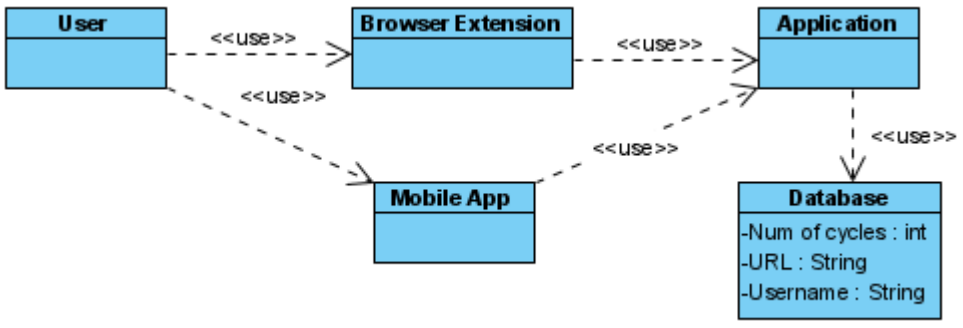
1.6. Assumptions and Dependencies

In order for the application to work as expected the following assumptions are made. To get started, the user's mobile phone must operate on the Android operating System, their web browser must be a chromium browser (i.e. Google Chrome; Microsoft Edge) and their personal computer's operating system has to be Windows in the case we develop a Windows App. If the user wishes to save his credentials across devices it is also expected that his devices have an internet connection.

2. System Architecture

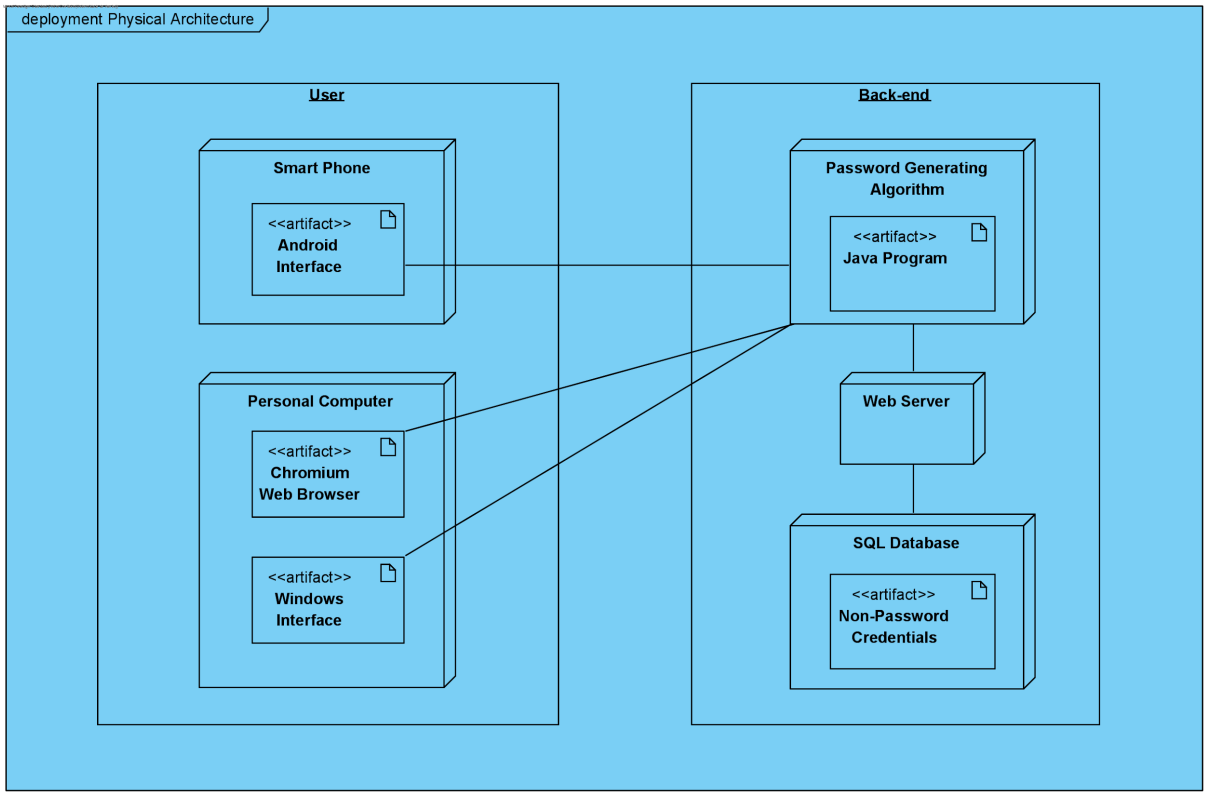
2.1. Domain Model

The domain model below describes the entities and relationships of the system. This system is very simple as requesting a generated password will only require our algorithm to get a set number of credentials, either entered by the user or entered by the app itself if they were previously saved and synchronized.



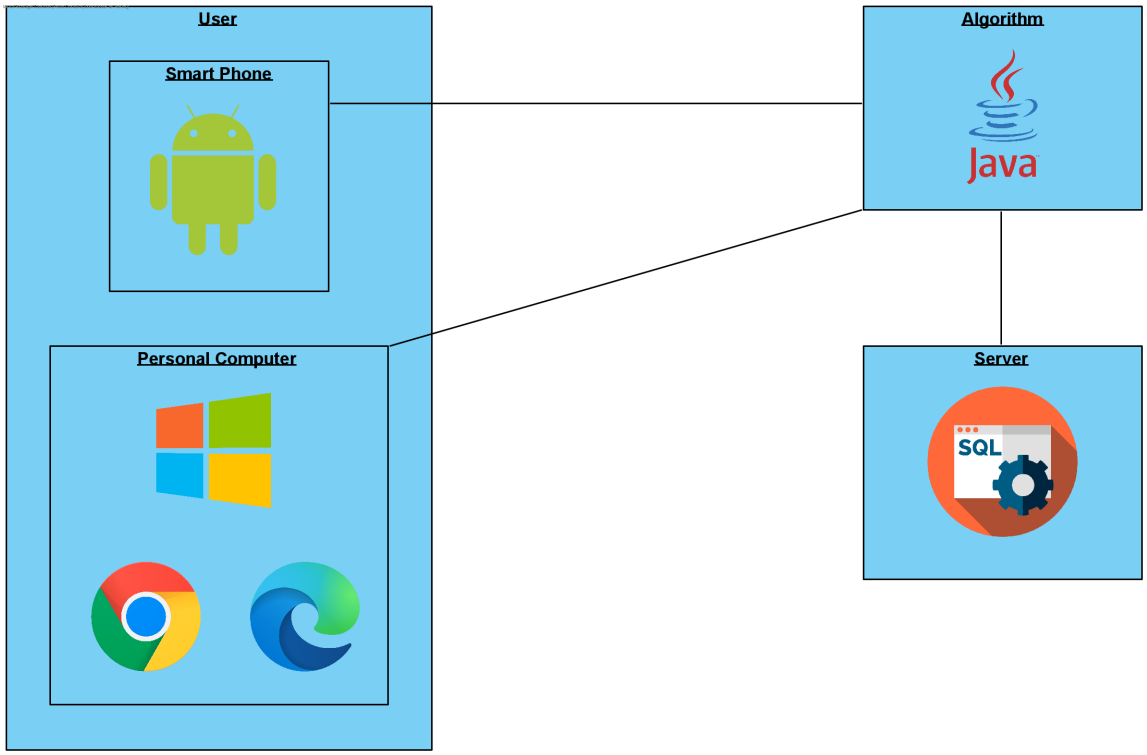
2.2. Physical Model

The following Diagram presents the physical architecture of the system.
The user's devices are the smartphone, with an Android interface and the personal computer, that should have a windows operating system (only if we make a Windows app) and a Chromium browser. Those devices will interact with our password generating algorithm, that was made in java, which will interact with a SQL database through a Web Server to retrieve URLs, usernames and counters if there's an internet connection and the user had them saved previously.



2.3. Technological Model

The technological model is another representation of our physical model where the technologies used within the service are clearly visible.



3. Contacts

- João Morais
  - 93288
  - antoniojoao10@ua.pt
- Miguel Ferreira
  - 93419
  - migueltf@ua.pt
- Pedro Coutinho
  - 93278
  - pmacoutinho@ua.pt
- Pedro Paixão
  - 73227
  - pedropaixao@ua.pt