Project BanQ

Project BanQ is a fictitious project and part of the Qualogy Polymer training. After completing Project BanQ you should be familiar with the fundamentals of WebComponents i.e. Custom Elements, Shadow DOM, HTML imports, and HTML Template; the Polymer library; and the Polymer Element Collection.

# Introduction

BanQ is a bank that wants to portray itself as the most modern bank on the market. To support this image, BanQ wants to create a new progressive web application (PWA). BanQ’s c-level executives decided that the application should leverage as much of what the different Web Components specifications have to offer. To ease development work its also decided that the Polymer Library and Polymer Element Collection will be used.

Initially, the application needs to have the following banking functionality: login/logout, accounts overview, transactions overview and creation, and an address book.

# Definition of Done

* **Polymer version 1.7 (Hybrid) is used.**
* **The Polymer HYBRID syntax is used unless explicitly stated differently. That is, mostly the syntax of Polymer 1 with some changes, see resources below.**
* The acceptance criteria are met.
* All components are tested, using Web Component Tester (WCT).
* All components are responsive i.e. it fits atleast desktop and mobile form factors. For example, on smaller screens the layout might be different or some content is omitted.
* The component is progressive i.e. it works for every user, regardless of browser choice.

# Resources

* <https://github.com/qafe/banq-restapi>
* <https://www.polymer-project.org/1.0/docs/devguide/feature-overview>
* <https://www.polymer-project.org/1.0/docs/api/Polymer.Base>
* <https://www.polymer-project.org/2.0/docs/upgrade> (Polymer Hybrid explained)
* <https://www.polymer-project.org/1.0/docs/tools/tests>
* <https://elements.polymer-project.org/>
* <https://elements.polymer-project.org/guides/flex-layout>

# Backlog

## Story 1: App Shell and Login

*Size: 4h*

### Description

BanQ likes to have the initial setup for the PWA. The PWA needs to be progressive and responsive. Also, a login view has to be available. After login, a temporary “Hello World” page is shown.

### Acceptance Criteria

* The PWA is Progressive i.e. it works for every user, regardless of browser choice.
* The PWA is Responsive i.e. it fits atleast desktop and mobile form factors. For example, on smaller screens the layout might be different or some content is omitted.
* The PWA is App-like i.e. it feels like an app to the user with app-style interactions and navigation.
* The PWA uses the Application Shell Architecture.
* When not logged in the user only sees the login view.
* When logged in the user sees a temporary “Hello World” page and is NOT able to navigate to the login page.
* The user is able to log-out.

### Resources

* <https://www.polymer-project.org/1.0/toolbox/server>
* <https://www.polymer-project.org/1.0/toolbox/app-layout>

## Story 2: Accounts and Routing

*Size: 8h*

### Description

BanQ would like to have an overview of accounts and transactions. An account has, among other things, a name, number, and balance. An account has transactions associated with it. Transactions have, among other things, a message, amount, from account, and to account. In addition, BanQ likes to have path routing, since it is descriptive, it allows the user to bookmark the current view, and leverages the browsers back and forward functionality.

### Acceptance Criteria

* Accounts can be viewed.
* Transactions can be viewed respectively for the accounts.
* Being at the login view is reflected in the URL path.
* Being at the accounts view is reflected in the URL path.
* Having an account selected is reflected in the URL path.

|  |  |
| --- | --- |
| info-icon-23.png | *We recommend using the following components from the Polymer team:*   * *“app-route” to implement routing* * *“app-storage” to persist data (e.g. the JWT token).* |

### Resources

* <https://www.polymer-project.org/1.0/toolbox/routing>
* <https://elements.polymer-project.org/elements/app-route>
* <https://elements.polymer-project.org/elements/app-storage>

## Story 3: Money Transfer

*Size: 8h*

### Description

How competitive is a banking application if it does not enable you to transfer money? Not that competitive. So, BanQ also likes to have this feature in their PWA. One can transfer money by creating a new transaction. A transaction has an amount, from account, to account, and message.

### Acceptance Criteria

* The user can transfer money from one of his accounts to another account.
* The transfer is reflected in the transactions overview.
* The transfer is reflected in the accounts overview.

## Story 4: Address Book

*Size: 8h*

### Description

A common and very convenient feature that most banking applications have is an address book. It lowers the threshold for users to use the application, since they do not have to have account information ready before initiating a money transfer. So, to stay competitive, BanQ also likes to have this feature in their new web application. An address consists of a name and a number.

### Acceptance Criteria

* There is an addresses overview.
* Addresses can be added, edited and removed.
* When creating a new transaction the user is able to auto fill fields by selecting an address.

## Story 5: Improve experience offline / on low quality networks

*Size: 4h*

### Description

Even the nicest and fastest web applications perform poorly on low quality networks or not at all if offline. Since BanQ is targeting all kinds of devices with their PWA, their users will encounter these kinds of network conditions. BanQ has had research done and if their application consistently behaves poorly, they will lose a significant number of customers. This is not acceptable, so to improve the experience during offline/low quality network conditions they want to use Service Workers to precache resources. In addition, they want their PWA to be discoverable such that e.g. it can be added to the home screen like a native application.

### Acceptance Criteria

* The experience in offline/low quality network conditions is improved.
* The different fragments of the application are lazily loaded.
* The PWA is discoverable and can be added to the home screen like a native application.

### Resources

* <https://www.polymer-project.org/1.0/toolbox/service-worker>
* <https://developers.google.com/web/fundamentals/getting-started/primers/service-workers>
* <https://github.com/GoogleChrome/sw-precache>