Fontys

Prepared By

Plamen Neshev

LITERATURE STUDY
HARDWARE & SOFTWARE

Table of Contents

Introduction

This research intends to gain insights on the most used framework for developing a POC and defining their advantages and disadvantages. The Project showcase POC can possible have information taken from the back end, which must be presented in the front end as text, photos, visuals, etc. Finding a good fit for presenting the information is of high importance.

Depending on what I manage during my prototype phase, this document can also serve towards my final advisory report.

Methodology

This research question isn't included in my Project Plan, but rather it derived from my main one.

What hardware would be the best at displaying projects by emphasizing the innovation at Fontys?

What software framework would best showcase and emphasize the innovative projects at Fontys?

Approach

To find an answer to my sub questions, I must conduct library research to find out:

- Best hardware that can emphasize Innovation Lab at Fontys
- Appropriate software for that hardware, that again emphasizes the Innovation at Innovation Lab

For hardware I will be looking into technologies that Fontys can provide like:

- Digital Screens
- Smartphone
- Augmented Reality (AR) Displays
- Video Walls
- Interactive Kiosk

For software framework I will be comparing between:

- Native
- Cross-Platform
- JS Libraries

Analysing Software & Hardware

The results of this research are categorized into two primary areas: hardware and software frameworks. This section provides a detailed evaluation of various hardware options and software frameworks that can effectively emphasize the innovative projects at Fontys. The insights gained from this analysis will inform the development of a Proof of Concept (POC) and contribute to the final advisory report.

Hardware

Digital Screens

Advantages:

- Attention-Grabbing
- Dynamical Content
- Easy Updates – real-time updates
- Flexible – administrators can decide on what content is on at what time.
- Cost-saving

Disadvantages:

- High Initial cost for setup – expensive cost of hardware & software
- Lacking Interactivity – lacking touch gestures
- Limited Visibillity – in an outdoor setting, weather can bring distractions
- Ad fatigue – although attention grabbing, screens are all around us to the point we ignore them

Smartphone

Advantages:

- App is convenient
- Can be used offline
- Helpful hardware features – camera, GPS,

Disadvantages

- Time consuming to create a mobile app
- Regular software updates, might hinder the application
- Lacking interaction between users, as everyone is looking on their phone

Augmented Reality (AR) Displays

Advantages:

- Immersive and Interactive:
- Competitive advantage: Not a lot of institutions are utilizing the AR technology
- Enhance Customer Experience: customers are interested in interacting with AR element

Disadvantages:

- Expensive
- Health and safety concerns - Several research studies suggest that overuse of AR technology negatively impacts your health, including eye issues, a state of mental dependence, and ear problems.

Video Walls

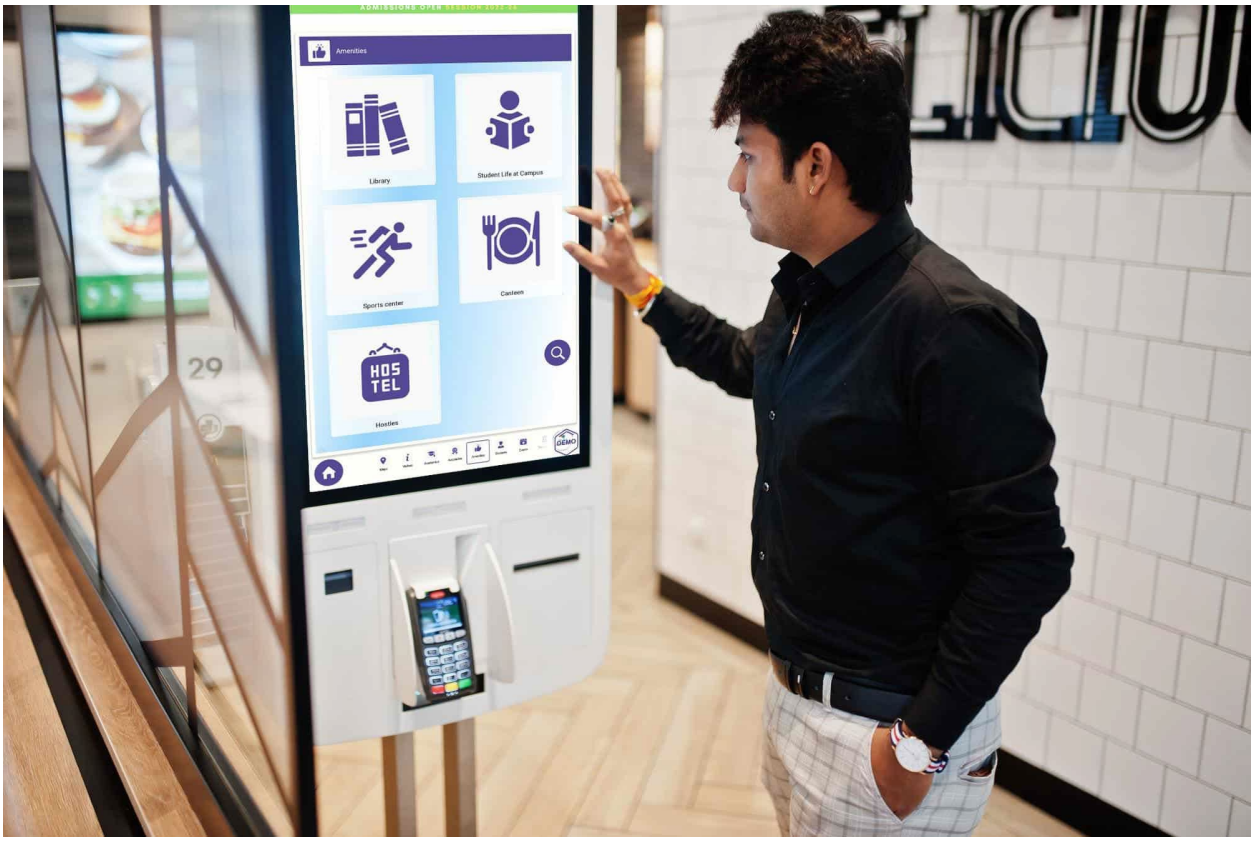
Advantages:

- No resolution limitations
- Processing power
- Looks Innovative
- Interactivity
- Long life-span

Disadvantages

- Expensive to setup
- Fragile
- If no touch gestuers are available, suffers from the same issue as normal digital screens

Interactive Kiosk



Advantages:

- Reduce Wait Time
- Information sharing
- Possibility of a personalized experience
- Interactive
- Touch-Free kiosk

Disadvantages:

- Limited customization
- Displacement of workers
- Initial Cost
- Lack of human interaction

Software Framework

Native

There are 2 native frameworks – Android & IOS, but for the purposes of this document I went with Android Development:

Native mobile development involves building apps for specific mobile operating systems like IOS or Android. Both Apple and Google provide app developers with their own development tools, interface elements and software development kits (SDKs).

Native mobile apps usually provide more user engagement than hybrid apps since they are built for the operating system. They also perform better than web-based apps and give access to different hardware on the phone, which might not be possible with a mobile browser.

Advantages:

-Dedicated Appstore – higher visibility than a web-based app

-Higher performance – built and optimized for a particular platform

-Features – native apps can immediately access the device's hardware such as GPS, camera and microphone

-Security – has security protocols built throughout the layers of the OS

Disadvantages:

-Time-to-market – higher time spent on separate coding of two apps

-High maintenance costs – maintaining separate apps for different platforms

-Development costs – difference in environments, programming, languages, etc.

-Platform – to develop it requires having both Android and Apple laptops

Cross-Platform

Flutter is a free and open-source mobile app development platform created by Google. It allows developers to create web, desktop and cross-platform apps that run on both Android and IOS. Flutter consists of two important parts:

- An SDK (Software Development Kit) - collection of tools created to help develop applications.
- A framework - collection of reusable UI elements (like buttons, text inputs, sliders, etc.)

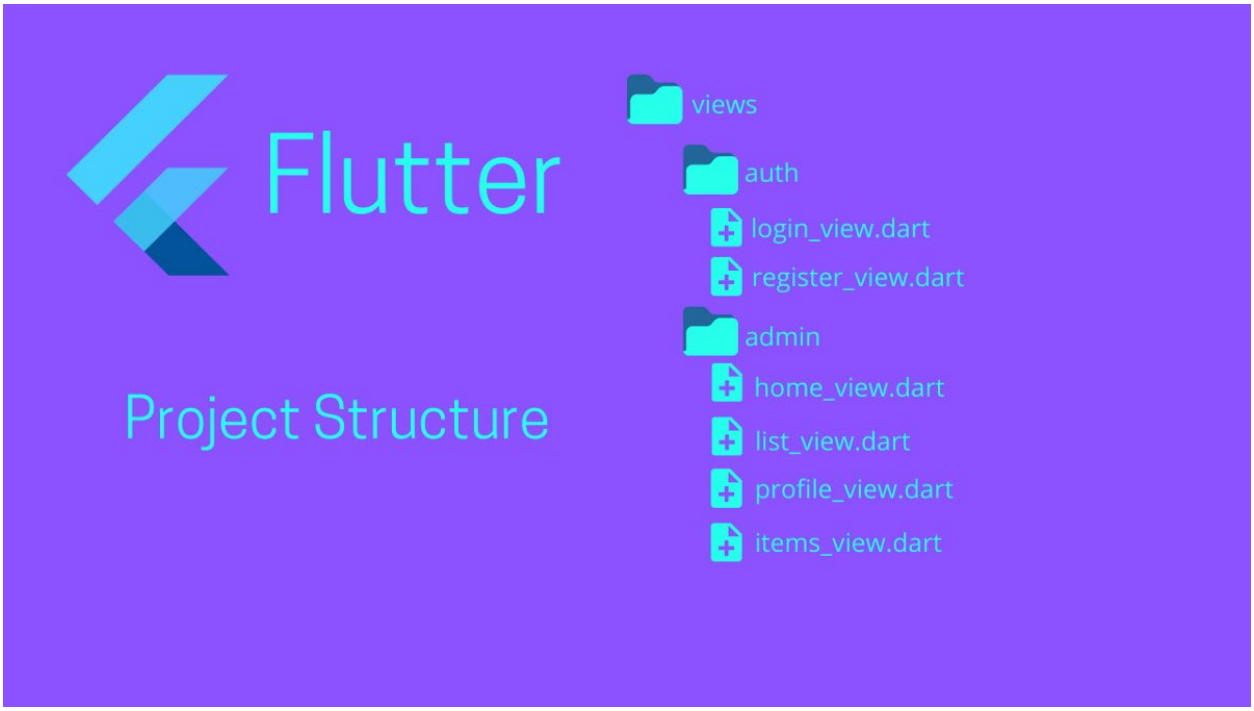
Developing with Flutter is with a language developed by Google – Dart. Dart focuses on front end-development, a typed Object programming language, mainly used to create mobile applications.

Advantages

-Cross-platform – Flutter allows having the same app on Android and in IOS

-Hot reload – update code and see in-real time updates

-Preset widgets – allow building beautiful UI with the help of built-in design-centric widgets



-Access to native features – Access geolocation, camera, etc. by using Objective-C, Swift or Java

Disadvantages

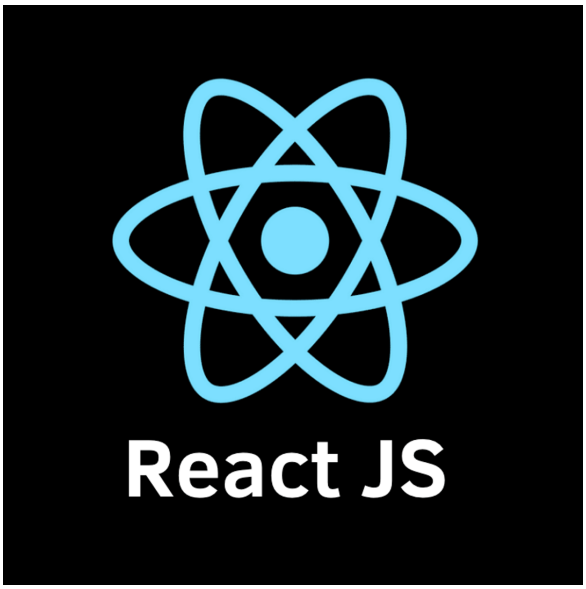
-Scarcity of 3rd party libraries – Because they are not many libraries for Flutter, if a functionality is missing, it is upon the developer to create said functionality

-Large Apps – Because of build-in widgets, the minimum app size exceeds 4MB, compared to native Kotlin (550KB)

- Limited Ecosystem – can't import JavaScript libraries into Flutter

JS Library

For the JS library framework, I went with React, as its one of the leading libraries in web-based development.



Advantages:

- Creating Components
- Scalable
- Flexible
- Easy to learn
- Good Performance
- Can handle Large applications

Disadvantages:

- Lack of Quality Documentation
- No standard way of developing aps
- JSX as barrier

Results

After carefully analysing this document and my findings, I have come to the following conclusions.

Hardware

While exploring different hardware options like digital screens, smartphones, AR displays, video walls, and interactive kiosks, I chose to focus on interactive kiosks due to their interactivity and user engagement. They are not passive like digital screens and video walls, unlike them they provide touch-based interface, that allow users to not only look at the content, but also engage with it. This is important for a project showcase, as it also emphasizes the innovative projects. They are efficient at showing detailed information, making them an effective solution for public space such as Fontys.

Software

Regarding the software framework, I chose ReactJS due to its capabilities in building dynamic and responsive user interfaces. It allows creation of interactive user experiences, which are important when it comes to engaging the user at the kiosk. ReactJS's component-based architecture promotes reusability making it easier to update and manage the application over time. Furthermore, ReactJS is supported by a large community and a has a lot of libraries and tools, securing continuous improvements and support.

Conclusion

In conclusion, the answer to my sub-questions is the following:

The combination of interactive kiosks for hardware and ReactJS for software provides a powerful platform for effectively showcasing and emphasizing the innovative projects at Fontys, ensuring a highly engaging and scalable solution.

References:

Digital Screens: https://www.quora.com/What-are-the-advantages-disadvantages-of-digital-screen-walls-for-advertisement

Pros & Cons of Digital Signage: https://www.risevision.com/blog/digital-signage-pros-and-cons

Advantages and Disadvantages of Mobile Apps: https://www.oppizi.com/post/advantages-and-disadvantages-of-mobile-apps/

The Advantages And Disadvantages Of A Mobile App for Your Business: https://objectiveit.com/blog/the-advantages-and-disadvantages-of-a-mobile-app/

Augmented Reality: https://bigohtech.com/advantages-and-disadvantages-of-ar-and-vr/#Advantages_of_AR

Benefits from using a videowall: https://vuwall.com/benefits-of-using-video-wall-and-how-they-work/

Video wall vs projector: https://dexonsystems.com/blog/video-wall-vs-projector

9 benefits of using a video wall: https://www.dataprojections.com/About/Insights/9-benefits-of-using-video-walls/

Advantages of an LED Wall: https://www.led-tek.de/en/2020/12/led-wall-advantages-video-wall/

Advantages and disadvantages of using a interactive kiosk: https://medium.com/@virtubox-io/what-are-the-advantages-and-disadvantages-of-kiosks-adcee4c32e5f

Advantages and Disadvantages of Self-Service Kiosk https://www.intuiface.com/blog/pros-and-cons-of-self-ordering-kiosk

Flutter

https://www.adservio.fr/post/what-is-flutter-and-what-are-its-advantages

https://www.freecodecamp.org/news/what-is-flutter-and-why-you-should-learn-it-in-2020/

What is native development?

https://mdevelopers.com/blog/what-is-a-native-mobile-app-development-

Native development vs Cross-platform

https://kotlinlang.org/docs/native-and-cross-platform.html

Pros and Cons of ReactJS

https://www.javatpoint.com/pros-and-cons-of-react

ReactJS — Advantages & Disadvantages

https://medium.com/@muhamed.salih/reactjs-advantages-disadvantages-16f479b3aa47