Joshua Rigsbee

WEB.289.0001

February 28, 2021

Virtua-Mall: Assessing Project Feasibility

The purpose of this document is to evaluate the development cycle of Virtua-Mall; an online shopping application designed to run on iOS devices. Evaluating these components should give us a better idea of what I can and cannot finish of this project in the given timeframe.

Technical Feasibility

I am required to release Virtua-Mall on the date of May 6th, 2021, which is little over a month from now as of writing. I will be honest, such a fast-approaching deadline has me slightly worried I might not finish the project in time. Finishing the project won’t be impossible; though, it will require a lot of time and responsibility. Ideally, I’d like to include most, if not all, of the features present in my original vision for the app.

Virtua-Mall will be made in Unity, a multi-purpose, multi-platform development framework that I have experience working in. I have decided to omit integration with Apple’s XCode IDE on account of how frustrating it can be to work with over a Remote Desktop connection. Likely, there will be times when I need to consult official documentation for Unity, specifically release 2018.4 of the Unity Editor (Unity Technologies). Furthermore, I am expected to incorporate external APIs for the purposes of user account management and information on items for sale. The UI demonstration for Virtua-Mall (made using Adobe XD) will continue to receive updates until I find it acceptable enough to replicate with Unity’s built-in components.

Making sure the explorable mall element is to my liking is the biggest priority next to integrating user accounts and customization features. A proof-of-concept should be ready within the next week or two, and I plan to share it with hopes of receiving feedback. I want to make sure this is all optimized to run on iOS devices, so I’m paying a great deal of attention to memory usage. Extreme caution will be taken into which APIs I decide to use, and they will be properly credited in Virtua-Mall’s documentation.

Following its release Virtua-Mall will receive occasional updates, which is to be expected of mobile applications with online functionality. Updates for this app will introduce new features, bug fixes, performance fixes, and additional quality-of-life improvements. These changes will be documented within the project’s GitHub page, as well as the app itself (via the ‘What’s New’ tab on the main menu).

Operational Assessment

I want to ensure that all personal information collected from Virtua-Mall’s userbase is handled responsibly. The following are examples of data that will be stored and where:

|  |  |
| --- | --- |
| Local Only | Local & Remote |
| * Real Name. * Location. * Shopping Cart Items. | * Nickname. * Password. * Email Address. * Date of Birth * Favorite Brands. * In-store items and prices. * Wish-listed Items. * Mall Customization. |

The stores in Virtua-Mall can be browsed offline; this is because item descriptions, prices, and pictures are saved to a cache after being downloaded. Cached data can be deleted from the settings menu in case it starts taking up too much space on the user’s device. In addition, any changes the user might make to their mall layout will be saved locally before being backed up to the cloud. Users will still need to enter their password regardless of being connected to a network. Local password data will be encrypted for security reasons.

Making sure the shopping platforms integrated into Virtua-Mall are being handled responsibly is just as imperative as safekeeping personal data. I’m currently looking into Amazon’s Product Advertising API (Amazon Web Services) to retrieve product information from their online storefront. Limiting the number of shopping websites/services to a select few that I can comfortably educate myself on will eliminate, if not greatly reduce, the risk of violating any guidelines these companies may have put in place. Eventually, I will add support for more storefronts once I have a betting understanding of these things.

Virtua-Mall is meant to be an interactive ‘front-end’ for a multitude of online shopping services that redirects users to the ones with the most reasonable prices and shipping. My original vision for this app was to be a fun experience for both children and adults but doing so requires compliance with Apple’s Review Guidelines. In the context of children, Apple prevents access to in-app-purchases unless some form of parental gate is instated (Apple).

Children using Virtua-Mall have special profiles created via their parent/guardian’s account. These profiles allow children to browse items, create wish lists, and customize their own section of the mall. A content filter ensures the items children can browse are age-appropriate, and they only see the suggested retail price for items. The parent user can view their child’s wish list to compare prices and makes purchases for them.

Another privilege parent users have is the ability to set parental controls to ensure safe and responsible usage of the app. This includes a daily time limit, adjustments to how strict the safety filters are, and the ability to disable layout customization. The mall layout editor might be difficult for some younger children to understand, which is why I’ve included this option. Any information associated with the child is tied to the parent account and is not accessible to any third parties.

There is an immense uncertainty, if not inevitability, of Virtua-Mall becoming the target of a devastating cyber-attack. Obviously, the solution is to implement a variety of precautionary measures to mitigate potential risks. The most important of which is making sure on-the-cloud user data is stored in secure servers that are periodically backed up to offline storage devices. If needed, users will be notified of data breaches and be asked to update their account passwords.

Schedule Feasibility

My plan is to have Virtua-Mall and its essential features ready to go at least 2 to 3 weeks before the May 6th deadline. I’ve broken the development process down on a week-by-week basis as follows:

Week 1 (2/28 – 3/6)

* Setting up the first-person mall experience with necessary coding, UI elements, and assets.
* Have a prototype ready for purpose of obtaining feedback.
* Start looking into potential APIs.

Week 2 (3/7 – 3/13)

* Make changes based on feedback and release 2nd prototype.
* Continue working on the UI, making sure it’s wired up correctly and complies with Apple’s guidelines.
* Make more assets for the mall environment (models, textures, sounds, etc.)
* Continue search for APIs.

Week 3 (3/14 – 3/20)

* Release 3rd prototype and revise based on feedback.
* Begin implementing external APIs.
* Start on mall layout editor and data save/load system.

Week 4 (3/21 – 3/27)

* Release 4th prototype and revise based on feedback.
* Continue API integration process; layout editor; data and account systems.
* Add more mall assets.

Week 5 (3/28 – 4/3)

* 5th prototype is released, feedback is received and used for further revision.
* Continue building previously mentioned components, in addition to more advanced user account features.
* Have project ready in alpha stage.

Week 6 (4/4 – 4/10)

* Thorough testing and revision of alpha build.
* Add more mall assets.
* Have first beta ready to go by next week.

Week 7 (4/11 – 4/17)

* 1st beta release.
* Test/revise ensure app is compliant with guidelines as it will be sent out for approval soon.
* If needed, add and tweak mall assets.

Week 8 (4/18 – 4/24)

* 2nd beta release.
* Revise app to release candidate status and submit for approval.
* Plan roadmap for future updates, start experimenting with possible new features.

Week 9 (4/25 – 5/1)

* Revise app and resubmit if need be.
* Continue working on potential updates.

Week 10 (5/2 – 5/6)

* Launch app on May 6th.
* Continue working on updates and revise based on initial public feedback.

There are two ways I can go about completing the Virtua-Mall project: I can either release the app fully-featured as I had envisioned it from the start, or with just the key features present (speed-to-market). The following assessment matrix shows the results of studying how feasible each approach is in several criteria:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Weighting | Fully Featured | Speed-To-Market |
| Technical Feasibility | 40% | App fulfills initial expectations, but some features might not work as intended since I wasn’t familiar with implementing them before.  Score: 60 | Having only the essential features at launch means the app will not meet my original vision.  A simpler start means the app is less likely to suffer from underlying issues.  Score: 75 |
| Operational Feasibility | 30% | There is potential for oversight of how user data is handled.  Legal issues likely to arise if a large selection of online storefronts is integrated.  Score: 40 | Limiting what users can do in the initial build means there is less likely to be issues with data handing.  Adding more customization options and online stores with each new update will provide long-term engagement to the userbase.  Score: 60 |
| Schedule Feasibility | 30% | Delivering an app with all the originally promised features would conflict with my work schedule and personal life.  Score: 20 | Despite being much simpler to complete, the project will still cut into my personal life. However, there are some aspects of developing it that I find mentally engaging, such as asset creation. I would not mind using my free time to work on textures and 3d modeling.  Score: 75 |
| Total Points | 100% | 120 | **210** |

# Referenced Materials

Amazon Web Services. *Product Advertising API 5.0 Documentation*. 12 November 2020. Document. 28 February 2021. <https://webservices.amazon.com/paapi5/documentation/>.

Apple. *App Store Review Guidelines 1.3 - Kids Category*. 1 February 2021. Document. 28 February 2021. <https://developer.apple.com/app-store/review/guidelines/#kids-category>.

Unity Technologies. *Unity User Manual (2018.4)*. 22 February 2021. Document. 26 February 2021. <https://docs.unity3d.com/2018.4/Documentation/Manual/index.html>.