Sinister Six

exposeum

Sinister Six

Olivier Brochu Dufour	26786049
Andrea Flores	26560598
Firas Krayem	25796571
Rahul Malik	29439951
Maxwell Mowbray	27005334
Chris Yazbek	27071663

Introduction

Team Sinister Six's project, Exposeum, aims to encapsulate the essence of a tailored, tour-guided visit to a museum into a mobile application. The project centers around the Musee Des Ondes Emile Berliner, formerly the RCA Victor factory, which is host to a rich history and hundreds of audio artifacts.

Currently, visitors to the museum must always be escorted by tour guides due to the complex layout of the site. However, a lack of personnel and funds greatly limits the amount of tours that can be undertaken at any given time. Exposeum offers to visitors an enriched and more autonomous experience.

Exposeum addresses this by offering a mobile application which museum visitors can download and use. At a high level, the app will allow users to follow a guided tour throughout the museum, as well as locate all exhibits in a free visit. Through audiovisual media, quizzes and other interactive features, Exposeum will offer a richer, more engaging museum-going experience.

The purpose of this document is as follows:

- Outline the requirements of the project
- Provide a detailed release plan from one iteration to the next
- Enumerate the risks at large and strategies to curtail their impact
- Showcase UI prototypes of the project as it develops
- Detail how features of the project will be tested

The intended audience of this technical document consists of the primary stakeholder Dr. Anja Borck, our course instructor Dr. Nikolaos Tsantalis, and our course tutors Jean Amirian and Davood Mazinanian. Furthermore, this document is intended for any future developers that may wish to research or continue the project after the term's completion.

While Exposeum is targeting the Musee des Ondes at present, it is being developed to be adaptable to other museums and venues.

1. Project Description

Exposeum is a Xamarin project slated for release on the Android platform. The following represent the essential tools and technologies used in the application.

iBeacon protocol (with Estimote beacons): iBeacons are low energy bluetooth devices that broadcast universal unique identifiers.

Estimote SDK for Android: The manufacturer of the iBeacons (Estimote) provides a library for Xamarin projects which allows us to search for and gather data from iBeacons.

SQLite: The internal database structure and its corresponding ORM (Object relational mapper) in an Android application consists of SQlite. With it, we will be able to easily store and retrieve persistent data.

Android Canvas: This built-in display method is similar to its HTML5 counterpart, and provides a method for displaying shapes and raster images.

NUnit (see 8. Testing Plan & Report): NUnit is a popular unit testing framework for the .Net / C# environment, and comes bundled with the Xamarin framework.

Xamarin UITest: Xamarin UITest is an automated mobile application testing framework which allows for system testing of the application's user interface.

Quickgraph: The project relies heavily on points of interest and traversable paths between them, therefore making a directed graph data structure the ideal choice as a model. Quickgraph happens to be the most used and recommended C# graph library for this purpose.

Android Support V4: This library allows us to use fragments as well as ViewPager and ImageViews, which would make off ultimately the splash page mentioned in our design.

Because Xamarin is a cross-platform mobile solution, it would be possible to create a version of the app for iOS device, provided the appropriate resources are made available.

Project Development Methodology

The Agile software development methodology is a popular approach to software development. Built into the core of this outlook are four key tenets: Individuals and Interactions, Working Software, Customer Collaboration, and Responding to Change.

The lifecycle of the project's development is structured into iterations, which are short timebox periods (2 weeks in our case), where selected tasks must be completed and a working prototype is produced and presented to the stakeholder.

We will follow the Agile methodology because its iterative nature allows for flaws or misunderstandings in the software requirements to be uncovered early on, given that each iteration is followed by a meeting with the stakeholder. Furthermore, retrospection about work completed and trouble encountered during a completed sprint allows for better cost and duration estimates.

The sprint schedule will be as follows:

Sprint	Date
0	11/01/2016-25/01/2016
1	26/01/2016-08/02/2016
2	09/02/2016-22/02/2016
3	23/02/2016-07/03/2016
4	08/03/2016-21/03/2016
5	22/03/2016-04/04/2016

2. Requirements

The following requirements were elicited from the product owner and have been turned into user stories approved by the product owner. ~~**16**~~ **18 user stories** have been elicited for a total of ~~**62**~~ **68 user story points**.

USP priority labels:

- High
- Medium
- Low

US-1	**As a Visitor, I want to specify my preferred language (english or french) at any time, so that I get information in a language I understand.**
USP	2
Priority	Medium
Description	

US-2	**As a Visitor, I want to view a list of up-to-date storylines available, so that I select the one that is most interesting to me.**
USP	3
Priority	Medium
Description	

US-3	**As a Visitor, I want to preview a selected storyline before starting it, so that I have an idea of what the story is about before I start it.**
USP	1
Priority	Medium
Description	The preview will include: number of points of interest, intended audience, estimated duration, etc.

US-5	**As a Visitor, I want to follow guided tours (storylines), so that so that I can get contextual information in the form of a narrative.**
USP	~~2~~ 5 **Increased USP:** We vastly underestimated the effort required for this user story.
Priority	High
Description	

US-6	**As a Visitor, I can engage a free tour mode of the building, so that I can visit all points of interest in an unrestricted way.**
USP	2
Priority	High
Description	

US-7	**As a Visitor, I can select any point of interest and view its summary when in free visit mode, so that I know if the POI is of any interest to me.**
USP	3
Priority	Medium
Description	

US-8	**As a Visitor, I can stop a storyline in progress and begin a new one so that I am not forced into completing a storyline if it does not interest me.**
USP	3
Priority	High
Description	

US-9	**As a visitor I want to receive push notifications when the app is not in focus so that no POIs go unnoticed.**
USP	3
Priority	Medium
Description	

US-10	**As a visitor I want to pause a storyline in progress and resume it at a later time so that I can complete a storyline at my own convenience.**
USP	~~1~~ 3 **USP Increased:** We underestimated the effort for this user story, it is now consistent with US-8 (start/stop)
Priority	High
Description	

US-11	**As a visitor I want to see which points of interest I have already visited so that I don't visit the same POI twice.**
USP	3
Priority	Medium
Description	

US-12	**As a visitor I want to view a progress map when in guided tour mode so that I know how many POIs are left in my guided tour.**
USP	5
Priority	High
Description	

US-13	**As a visitor I want to receive full contextual information about a point of interest in my proximity so that I get more educated about each POI I visit.**
USP	8
Priority	Medium
Description	

US-14	**As a visitor I want to view the entire map of every floor with all points of interest when in free visit mode so that I can choose which POI to visit.**
USP	8
Priority	Medium
Description	

US-16	**As a visitor I want to scan QR codes so that I have more information about certain POIs.**
USP	3
Priority	Low **Priority lowered:** Feature was subject to much debate (still unresolved) and stakeholder does not consider it a core feature.
Description	

US-18	**As a visitor, once I have finished a storyline, I want to be directed to the closest exist or back to the venue start point.**

USP	13
Priority	Low **Priority lowered:** Stakeholder indicated this feature is of low concern.
Description	

US-19	**As a user, I want to receive an intro to the app so that I can immediately familiarize myself with its use.** **Added to S2:** Stakeholder really loved the idea during the presentation.
USP	3
Priority	Medium
Description	

US-20	**As a user I want to choose between Guided tour and an Explorer tour so that I can pick between the app's two main modes.** **Added to S2:** Error of omission in past sprints (it was an obvious story that was needed)
USP	3
Priority	Medium
Description	

US-21	**As a visitor I** want to be able to reset the application to it's original state

	Added to S3: Error of omission in past sprints (it was an obvious story that was needed)
USP	3
Priority	Medium
Description	

US-25	**As a user I want to be able to skip a POI and continue the current Guided Tour.** **Newly added to S4**: added by stakeholder after presentation of Sprint 3
USP	5
Priority	Medium
Description	The user will be able to choose between going back to the missed POI or continuing the current storyline when a POI is skipped.

US-26	**As a visitor, when resuming a Guided Tour, I want to be directed from closest POI to the last visited POI.** **Newly added to S4:** added by stakeholder after presentation of Sprint 3
USP	5
Priority	Medium
Description	When user resumes the storyline in progress, once he hits a range of a POI, it will show him the closest path to the last POI visited in the current storyline.

Backlog

ID	Name	USP	Priority
US-1	As a Visitor, I want to specify my preferred language (english or french) at any time.	2	Medium
US-2	As a Visitor, I can retrieve a list of up-to-date storylines.	3	Medium
US-3	As a Visitor, I want to preview a selected storyline before starting it.	1	Medium
~~US-4~~	~~As a Visitor, I want to specify my age group (child or adult).~~	~~1~~	~~Medium~~
US-5	As a Visitor, I want to follow guided tours (storylines).	5	High
US-6	As a Visitor, I can engage a free tour mode of the building.	2	High
US-7	As a Visitor, I can select any point of interest and view its summary when in free visit mode.	3	Medium
US-8	As a Visitor, I can stop a storyline in progress and begin a new one.	3	High
US-9	As a visitor I want to receive push notifications when the app is not on focus.	3	Medium
US-10	As a visitor I want to pause a storyline in progress and resume it at a later time.	3	High
US-11	As a visitor I want to see which points of interest I have already visited.	3	Medium
US-12	As a visitor I want to view a progress map when in guided tour mode.	5	High
US-13	As a visitor I want to receive full contextual information about a point of interest in my proximity.	8	Medium
US-14	As a visitor I want to view the entire map of every floor with all points of interest.	8	Medium
~~US-15~~	~~As a visitor I want to hear ambient or audio in between POIs.~~	~~8~~	~~High~~
US-16	As a visitor I want to scan QR codes.	3	Medium Low
~~US-17~~	~~As a visitor I want to be presented with a game/quiz during my guided tour.~~	~~2~~	~~High~~
US-18	~~As a visitor I want directions for the shortest path between two POIs.~~. As a visitor, once I have finished a storyline, I want to be directed to the closest exist or back to the venue start point.	13	~~High~~ Low
US-19	As a user, I want to receive an intro to the app.	3	Medium
US-20	As a user I want to choose between Guided tour and an Explorer tour.	3	Medium
US-21	As a user i want to be able to reset the application to it's original state	3	Medium
US-22	As a user I want to be able to skip a POI and continue the current Guided Tour.	5	Medium
US-25	As a visitor, when resuming a Guided Tour, I want to be directed from closest POI to the last visited POI.	5	Medium
Total		84	

***added in current sprint**

Internal Backlog

This backlog reflects work to be done which does not directly represent business value to our stakeholder.

ID	Name	USP	Priority
US-23	As a developer, I want to improve the structure of the application code.	3	Medium
US-24	As a developer, I want to improve the structure of the internal app database.	3	High
US-26	As a developer, I want to acquire and parse map JSON data.	5	High
US-27	As a developer, I want to clean up the UI and make it more user-friendly.	5	Medium
US-28	As a developer, I want to implement dependency injection to reduce coupling and increase cohesion.	5	Low
Total		21	

3. Release Planning

Sprint 5 Summary

Sprint 5 consisted of a mixture of different user stories:

- Initially very high risk story that had to be pushed back to late development (JSON)
- Crowd pleasing refinement (UI)
- Late-game stakeholder wants (skip POI, resume with shortest path)
- Internal structural improvements (dependency injection and refactorings)

The result of this eclectic sprint was the maturation of the system into one that is near release-ready.

Story ID		USP	Status
US-23	As a developer, I want to improve the structure of the application code.	3	**DONE**
US-22	As a user I want to be able to skip a POI and continue the current Guided Tour	5	**DONE**
US-25	As a visitor, when resuming a Guided Tour, I want to be directed from closest POI to the last visited POI.	5	**DONE**
US-26	As a developer, I want to acquire and parse map JSON data.	5	**DONE**
US-27	As a developer, I want to clean up the UI and make it more user-friendly.	5	**DONE**
US-28	As a developer, I want to implement dependency injection to reduce coupling and increase cohesion.	3	**DONE**
Total		26	

Project velocity after 5 sprints: 20.8 (+1.3 from S4)

Sprint 5 Burndown chart

This is the user story points burndown chart. Despite appearing incomplete, all user stories were in fact completed (some were left “ToDo” by accident).

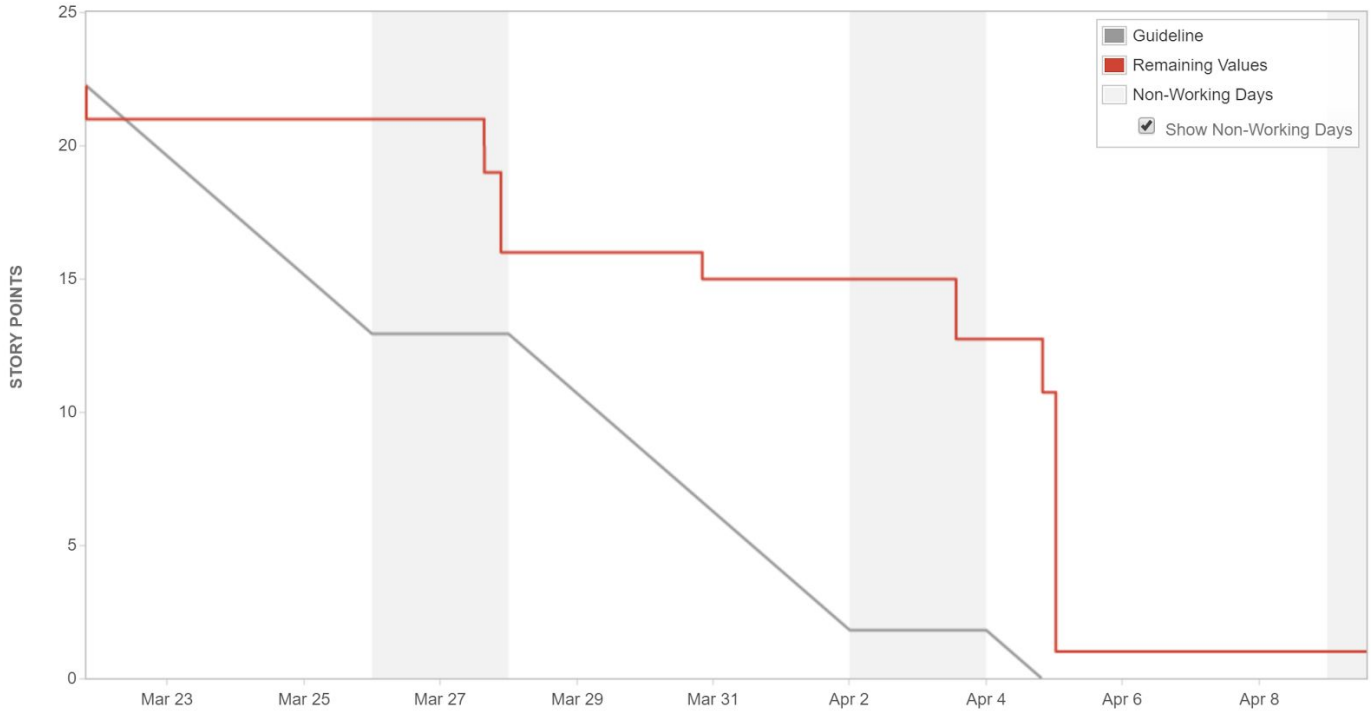


Figure 1 - Sprint 5 Burndown Chart

Sprint 5 Retrospective

Keep doing:

- Weekly complete meetings: Since the entire team can seldom meet due to work and class schedules, these weekly meetings are extremely helpful in unifying the team's vision of the project, its requirements and the sprint document.
- Biweekly sub-team meetings: The decision to conduct sub-team meetings was a good one, this allows for more frequent meetings not subject to the entire team's schedule constraints.
- Daily 15-min scrum-style meetings: It is easy to get distracted by other classes, projects and life outside of school, so having a quick daily meeting to ask how things are going in regards to tasks related to Exposeum is vitally important.
- Sub-team assignments: Assigning more than one member for features helps to promote involvement and engagement, plus pair programming is more fun.
- Mandatory teamwork hours: We decided to dedicate a collective hour of work with the whole team, this resulted in very efficient progress in both the code and document with minimized communication delay time.

Start doing:

- In-depth overview of the upcoming Sprint: It helps to get a breakdown of sprint requirements as early as possible, and delegate tasks to sub-teams and their members.
- Transfer of Knowledge: So the entire team benefits from the effort and time spent learning a tool or a work around.
- Reporting tools: Learn how to better use them to benefit the development process and the report / tracking.
- Learning metric tools ASAP**:** Given the amount of effort spent attempting to make SonarQube and DotCover work with a Xamarin project, we realized that perhaps we waited too long before learning how to use new tools.
- Diligent linking of user story status changes in Jira / Git: Doing so will permit us to quickly determine where, when and why a commit was made and what it affected in overall progress.

Stop doing:

- Documentation: Under-estimating the workload behind the documentation.
- Code Focus: Attributing a lot of time for development instead of dividing it equally with testing and reporting.
- Unilaterally changing dev branch without pull requests**:** Some "emergency" features needed to be propagated immediately but this still represents poor practice.
- Not updating documentation / diagrams after changing code: Doing little changes to documentation as we change code divides the responsibility and load among all team members (rather one one person tackling the task nearing end of sprint).

4. Architecture

Domain Model

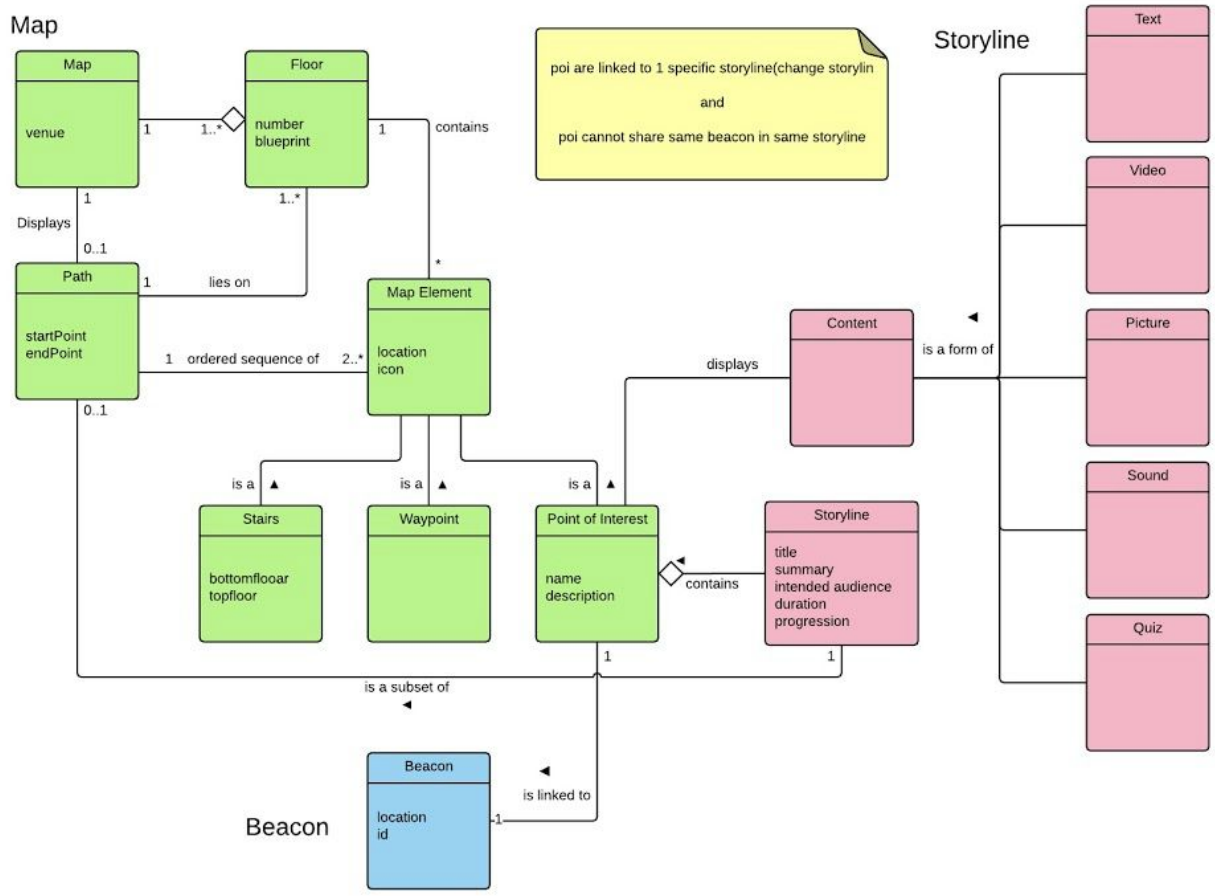


Figure 2 - Domain Model

Domain Model overview:

Our domain diagram consists of three distinct conceptual packages. The Map package concerns itself with the modeling of the venue or building into a map, its constituent floor plans and map elements (POIs, stairs, etc.) located within. Paths as conceptualized are ordered sequences of map elements which allow visitors to go from the start of the path to its end.

The Storyline package relates to the narrative intended to be shared with a visitor undergoing a walking tour of the museum. A storyline is an ordered sequence of Points of Interests that pertain to the narrative. A POI contains story content to be displayed to the visitor.

Albeit small, the Beacon package is an integral part of our domain. It is solely responsible for linking the real world beacon hardware with our conceptual interpretation of the museum and storyline paradigm.

Class Diagram

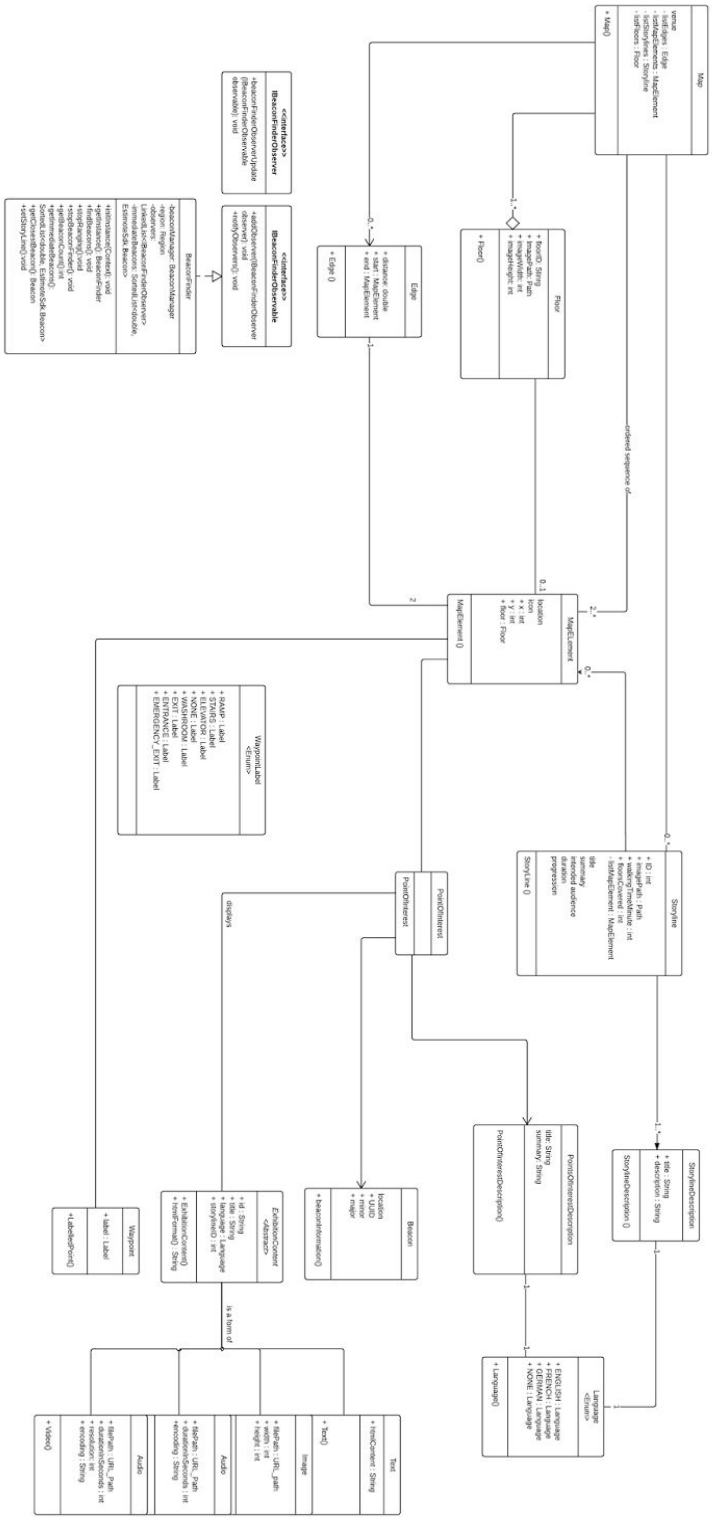


Figure 3 - Exposeum Class Diagram

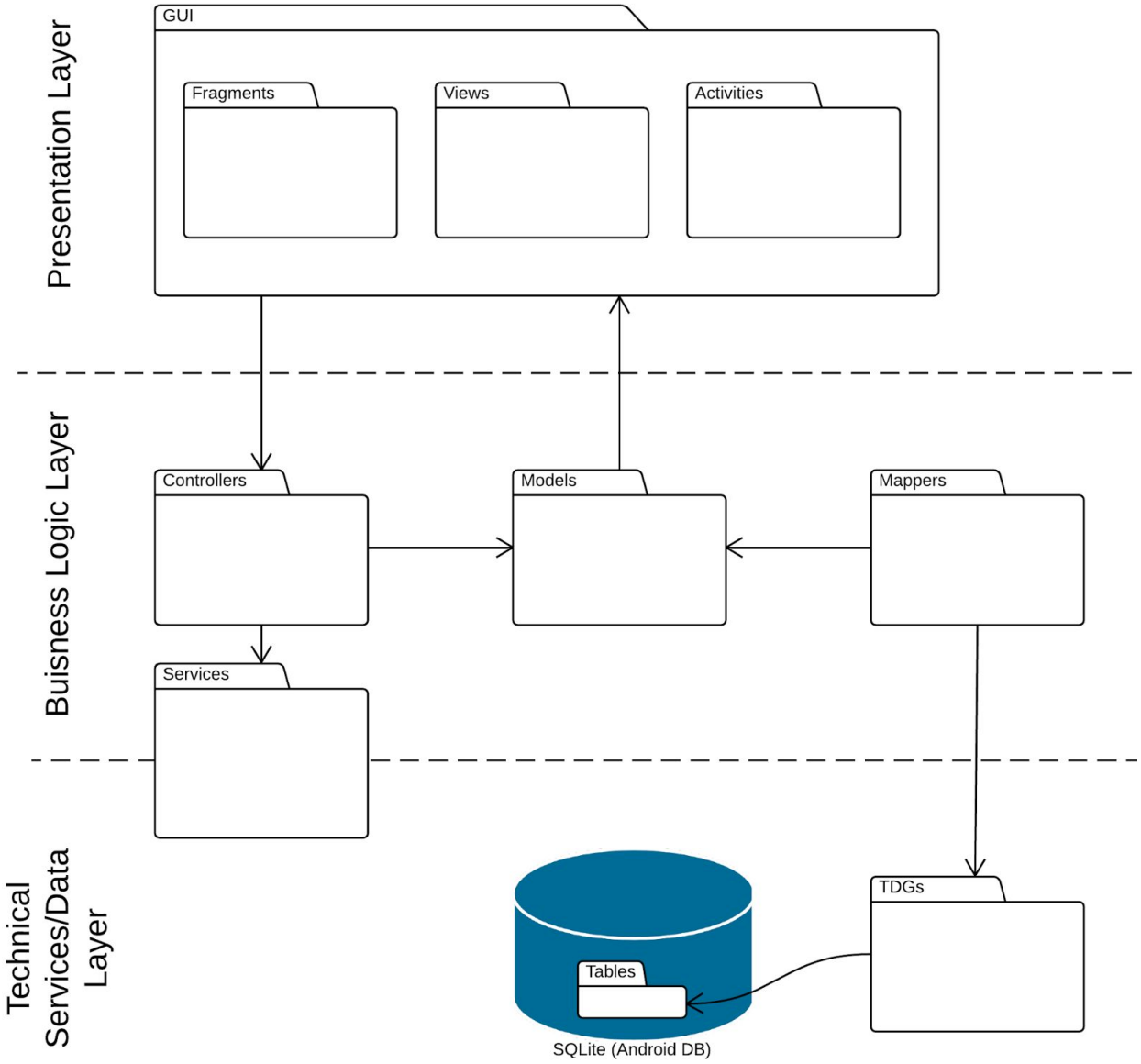


Figure 4 - Package Class Diagram

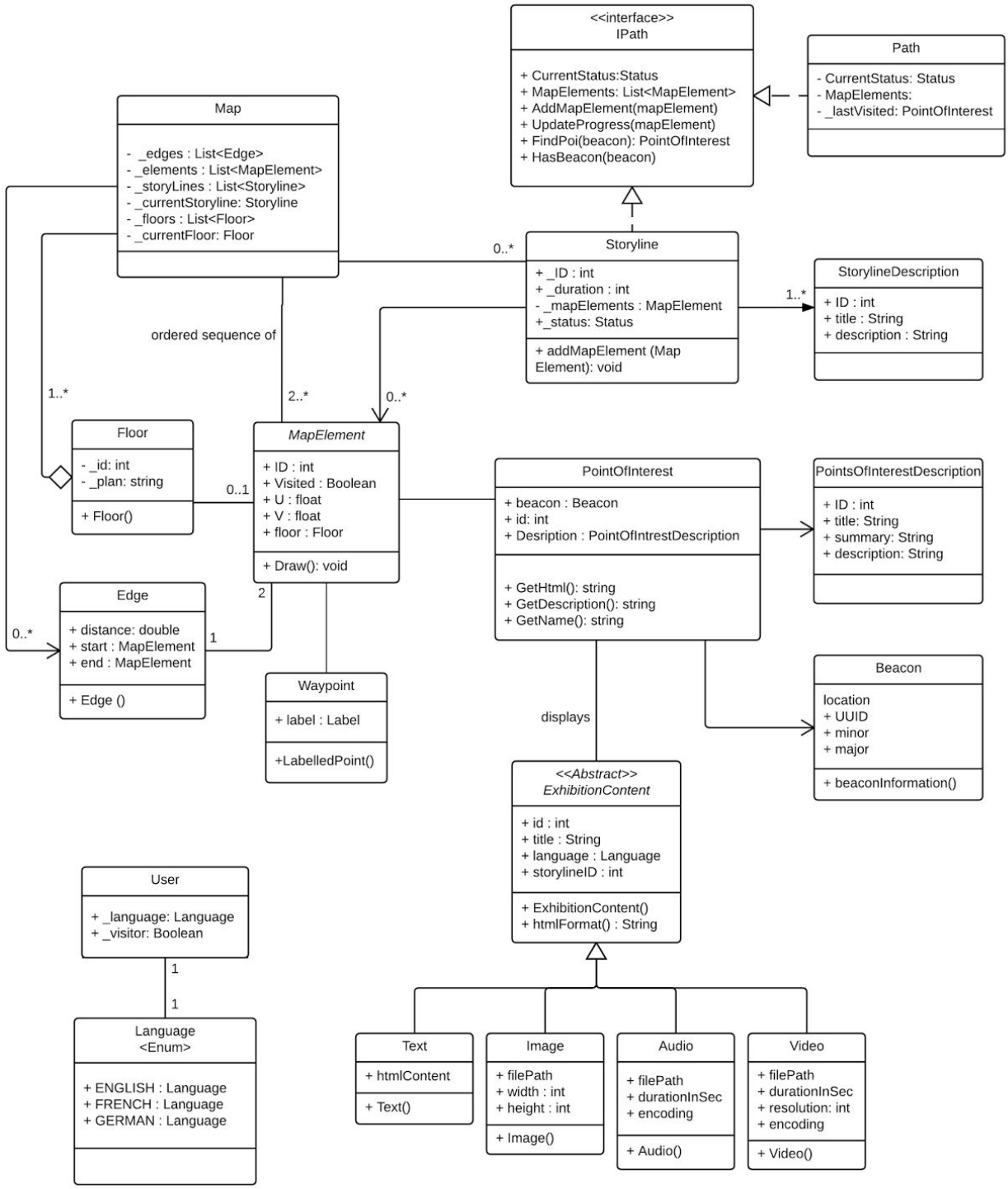


Figure 5 - Models Package Class Diagram

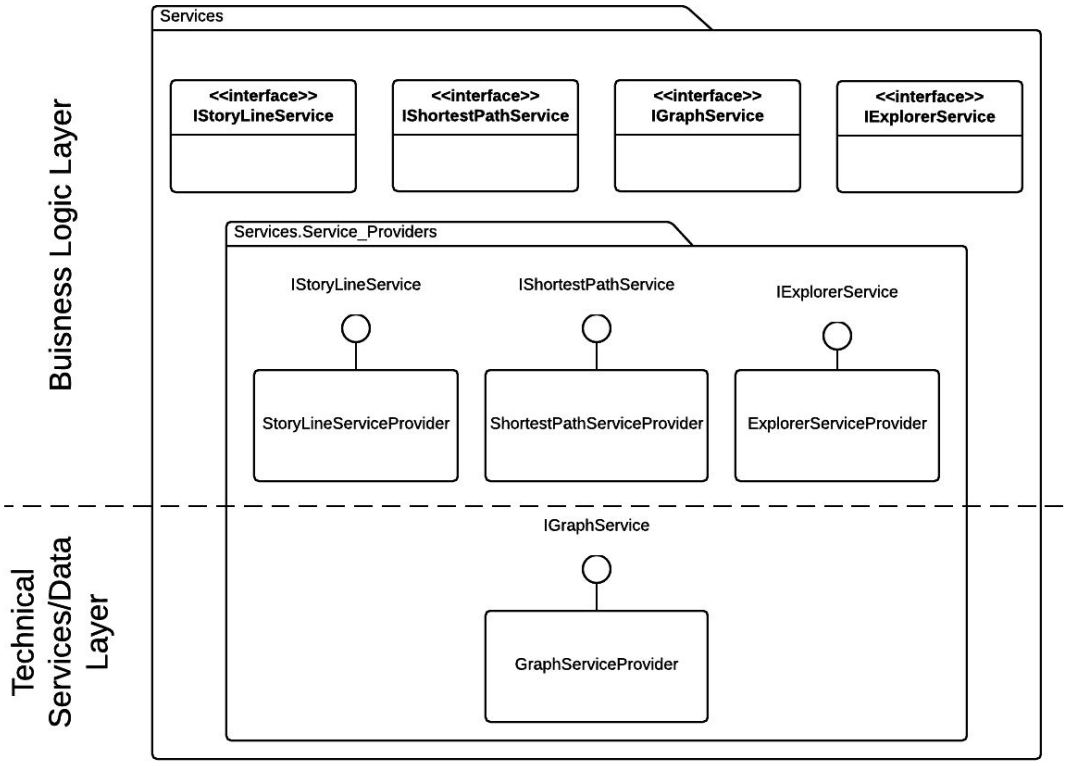


Figure 6 - Services Package Class Diagram

Class Diagram overview

The representational gap between our domain model and actual class diagram is relatively small, with the primary differences due in part to purely object-oriented programming concepts.

Team Sinister Six did not need to modify our class diagram much since our original was very close to the final diagram derived from consensus with the other teams and Professor Tsantalis. The core and common design is represented above with additions made for our needs.

The BeaconFinder class implements the IBeaconFinderObservable interface, and is used to detect and enumerate nearby detected beacons through the observer pattern.

Software and Architectural Patterns

Dependency Inversion Principle (DIP)

The goal of the Dependency Inversion principle is to prevent tight coupling in software projects. It states that high level modules should never depend on low level modules, and that both should instead depend on abstractions (interfaces). Using abstractions over concrete implementations whenever possible has many benefits: factory classes can provide any implementation of the interface specified at run-time in a configuration file, the abstraction can be faked by a stub class in unit tests, etc. Additionally, adhering to DIP enables the use of a dependency injection framework and dependency containers.

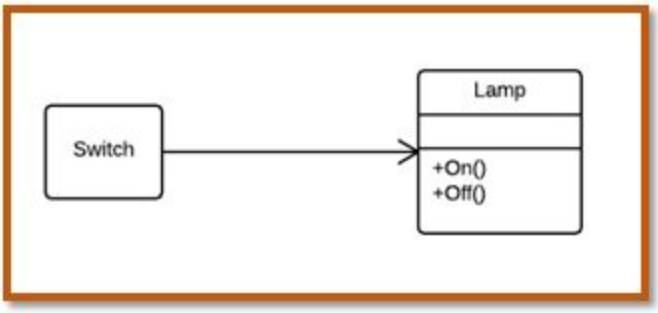


Figure 7 - Switch depending on concrete Lamp

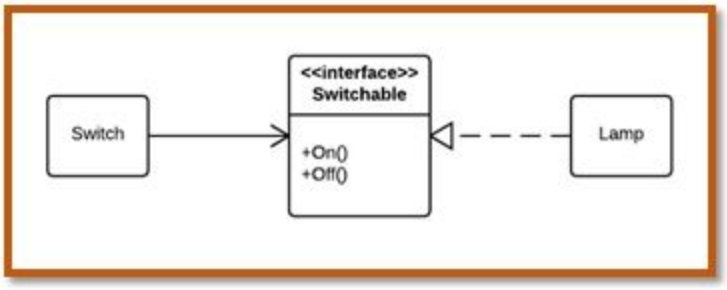


Figure 8 - Switch depending on abstraction (it can now operate any switchable entity)

In Exposeum, we used the Ninject dependency injection framework (found on Nuget). All Services found in Exposeum were made compliant to the dependency inversion principle in Sprint 5.

Model-View-Controller (MVC)

MVC is a commonly-used architectural design patterns. The principle is that the job of a view is merely to represent data contained in models. When a user interacts with a view, a command is triggered in the corresponding controller, which is where the logic lies, and is typically where models are manipulated. As depicted in Figure 4, a package containing controllers was created, representing the bulk of the business logic of Exposeum.

Table Data Gateways

Exposeum uses Table Data Gateways (TDG) and Data Mappers. The TDGs are in charge of manipulating the database's tables and the Data Mappers are in charge of moving data between objects and tables while keeping them independent.

Component Diagram

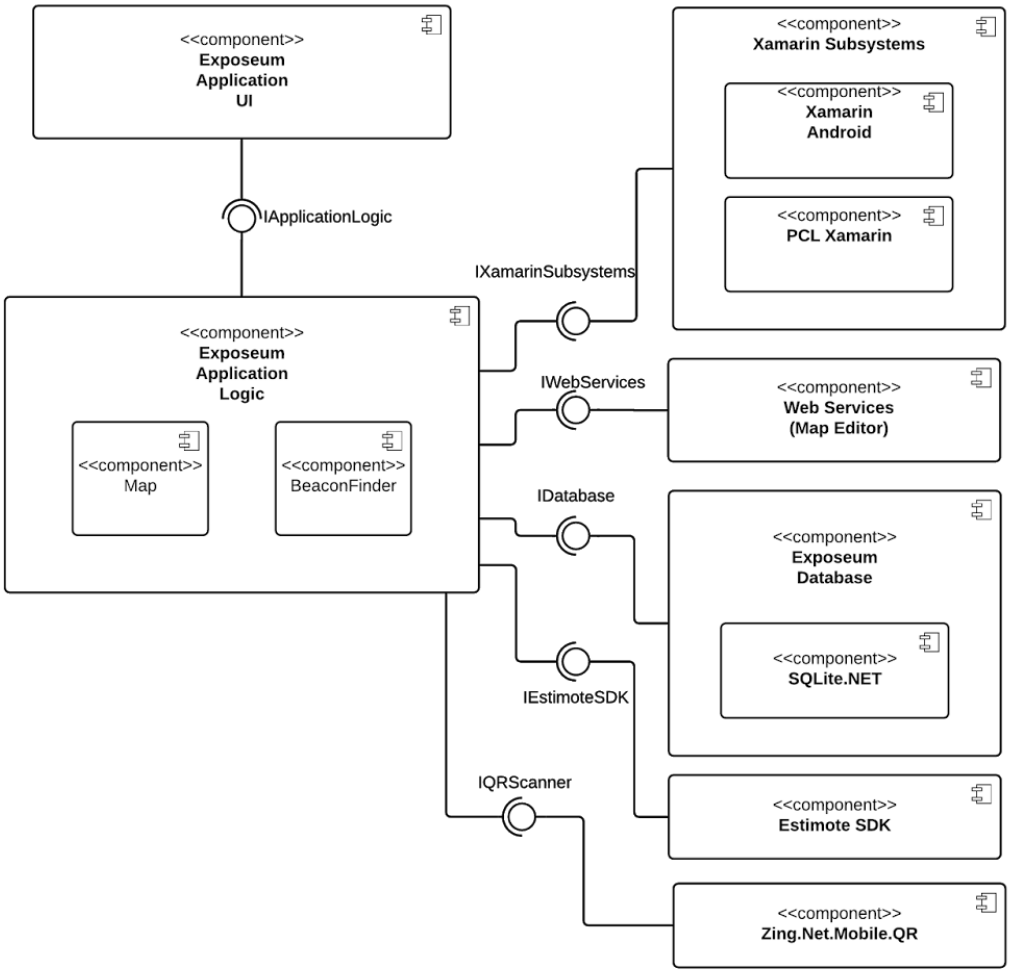


Figure 9 - Component Diagram

The Exposeum Application UI component is in charge of displaying visual elements and contextual information. It receives data through the IApplicationLogic interface from the Exposeum Application Logic component. This component is in charge of the core computation and application specific logic such as finding the beacons (done through the BeaconFinder component) and drawing the map (done through the Map component). The Exposeum Application Logic is dependent of the following components:

- Xamarin Subsystems component: low-level compatibility of Xamarin C# with native Java Android layer.
- Web Services component: retrieval of external map and beacon data destined for local persistence.
- Exposeum Database component: Allow for data persistence and object serialization and deserialization.

- Estimote SDK component: interfacing with physical iBeacons via Bluetooth.

5. Risk Management

The table below identifies the risks, their probability & their impact as well as the strategy that Sinister Six will adopt to tackle each risk. The risk is assessed in qualitative manner following the criteria below:

High: Extremely likely to occur / Represents a High adverse impact
Significant: Very likely to occur / Represents a significant adverse impact
Moderate: Somewhat likely to occur / Represents a moderate adverse impact
Low: Unlikely to occur / Represents a low adverse impact

Impact	**High**	PE03	PE04, RE01, RE02		TE02, PE02
	Significant			PE01	
	Moderate		TL03, TL04		
	Low	RE03			TE01, TL01, TL02
		Low	**Moderate**	**Significant**	**High**
		Probability			

* Entries in **bold** are updates/additions from the previous sprint to the current sprint

Risk Assessment				Risk Management	
Risk ID	**Description**	**Probability**	**Impact**	**Resolved In Sprint**	**Strategy & Effectiveness**
Technology Risks					
TE01	Team Sinister Six has no experience with the use of beacon technology (such as Estimote iBeacons).	High	Low	0	**The team has purchased an extra set of Estimotes with which to practice prior to Sprint 0.** **Strategy:** Mitigation **Effectiveness:** High
TE02	The format of the map data is yet unknown and team Sinister Six must wait for consensus to be reached.	High	High	2	**Team leads of several teams met with the professor and reached a consensus on the final JSON schema.** **Strategy:** Elimination **Effectiveness:** High
Tools Risks					
TL01	Some members of team Sinister Six are unfamiliar with Visual Studio (one IDE option for Xamarin development).	High	Low	0	**The team has practiced with visual studio, installed helpful extensions (ReSharper) and has subscribed to an online tutorial website using our academic accounts (PluralSight)** **Strategy:** Mitigation **Effectiveness:** High
TL02	All members of team Sinister Six are unfamiliar with Xamarin Studio (the other IDE option for	High	Low	0	**The team has agreed to use Visual Studio for development.** **Strategy:** Avoidance **Effectiveness:** High

	Xamarin development)				
TL03	Using both Visual Studio and Xamarin Studio for app development concurrently may lead to incompatibility and/or communication issues amongst team members	Moderate	Moderate	0	**The team has agreed to use Visual Studio for development.** **Strategy:** Mitigation **Effectiveness:** High
TL04	The Estimote (iBeacon) SDK is closed source, external tool. The SDK contains bugs and makes unit testing more difficult, which may lead to delays and blocks.	Moderate	Moderate	3	**We worked around some of the testing issues (adding try / catches and substituting dummy resources).** **Strategy:** Acceptance **Effectiveness:** Moderate. Some issues still persist, Xamarin is proving to be a hurdle in the way of proper testing.
People Risks					
PE01	Personnel conflict / conflicting personalities	Significant	Significant	1	**Implementing open dialogue between team members and ensuring the hierarchical structure of the team is always respected.** **Strategy:** Mitigation **Effectiveness:** High
PE02	Scheduling conflicts leading to limited time slots in which the team can meet to work in tandem.	High	High	1	**Work around everyone's schedules, using different scheduling technologies like Doodle to help find a common free slot to all team**

					members. **Update: We have split into subteams where each subteam member is able to meet weekly.** **Strategy:** Acceptance **Effectiveness:** Low (the problem persists and we must accept its impact).
PE03	Team member dropping class.	Low	High	0	**Redistribute the workload between remaining team members.** **Strategy:** Acceptance **Effectiveness:** High
PE04	Sub-par team member performance as compared to other team members.	Moderate	High	0	**Team leader addressing team members performance on a weekly basis.** **Strategy:** Avoidance **Effectiveness:** Moderate
Requirement Risks					
RE01	The stakeholder's experience with technology and vision of the product is not firmly set thereby leading to	Moderate	High	1	**Personal in-depth interview conducted with the primary stakeholder clarified and solidified much of the remaining unclear requirements.**

	unclear and volatile requirements.				**Strategy:** Mitigation **Effectiveness:** High
RE02	Familiarity with some technologies (QR, AR, etc.) is low amongst the team members and the stakeholder	Moderate	High	2	**Meeting with other team leads and the professor helped us dramatically reduce the priority of the QR/AR feature, almost to the point of total removal. The feature was found to be too vague and too out of scope for the vision of the project.** **Strategy:** ~~Mitigation~~ Elimination **Effectiveness:** High
RE03	Attempts to refactor and introduce better architecture in the system is yielding intellectual debates and learning curves (for example, with dependency injection)	Low	Low	5	**Refactoring slowly and methodically reduced risk impact and duration.** **Strategy:** Acceptance **Effectiveness:** High, the risk impact was low and easily managed by following good refactoring practices (small changes followed by testing)

Sprint Risk Overview:

Resolved risks

RE03: This late risk did not impact us much and was easily managed. Refactorings were successful and relatively easy to implement.
PE03: The probability of a team member dropping the course at this point is of no consequence and will not deter the progress of the project.

Updated Risks

None

Added Risks

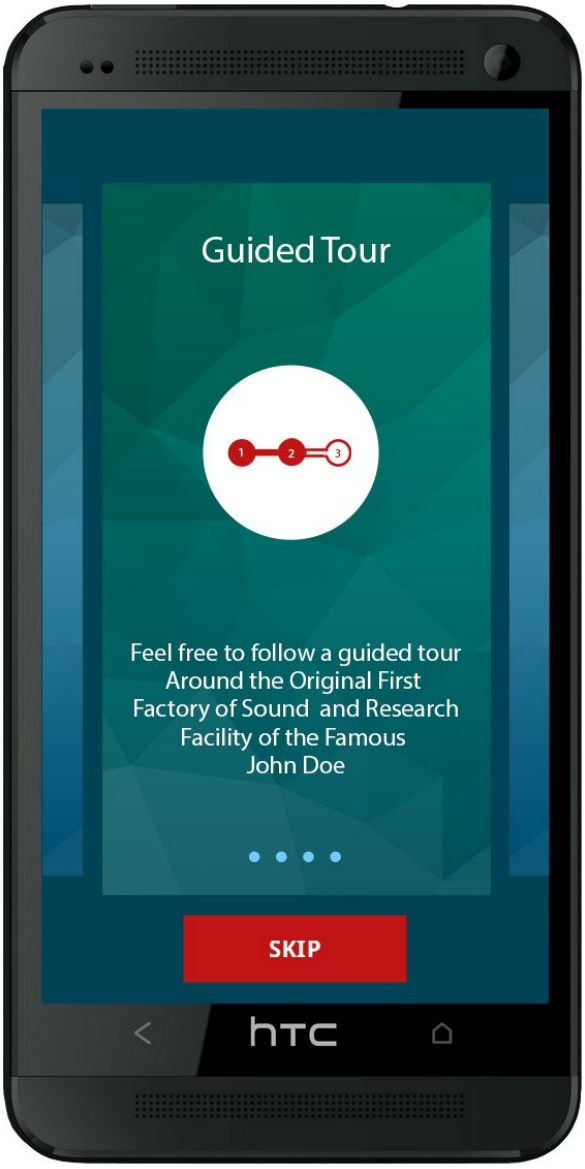
None

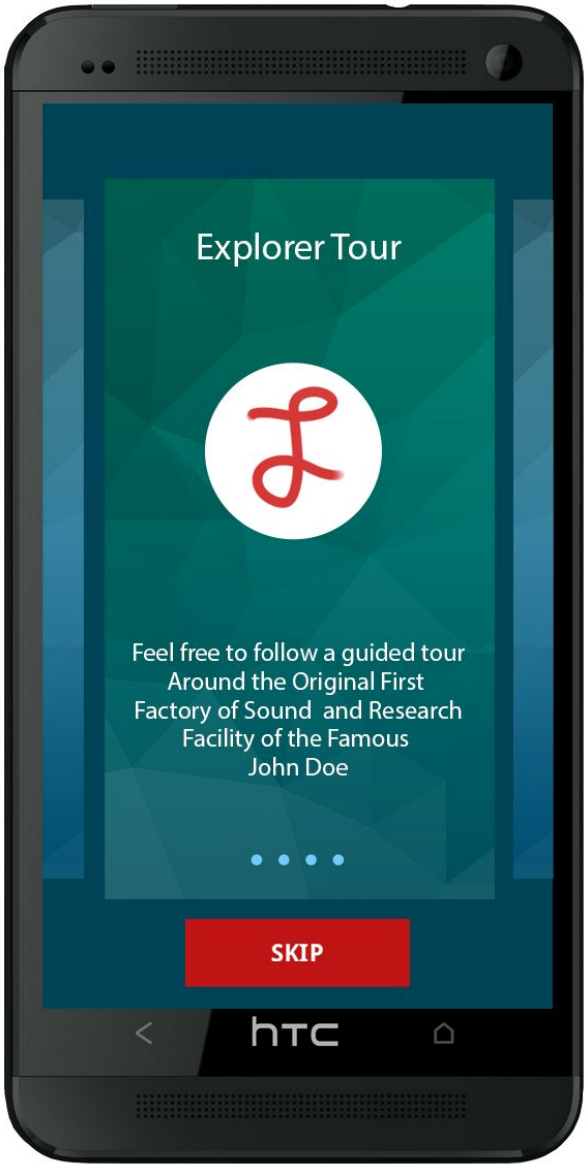
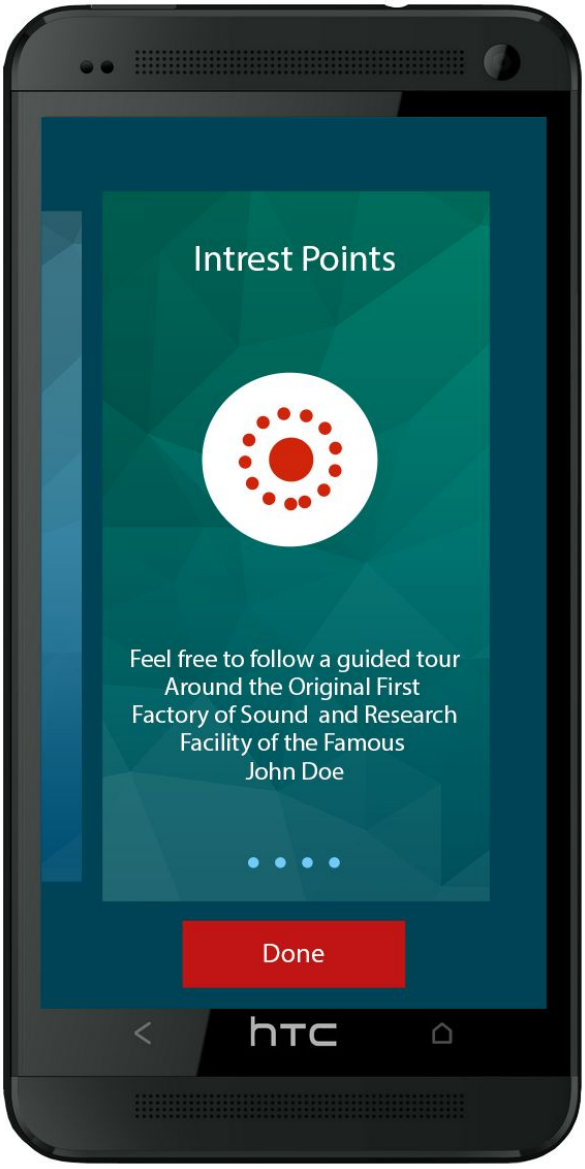
6. User Interface Design

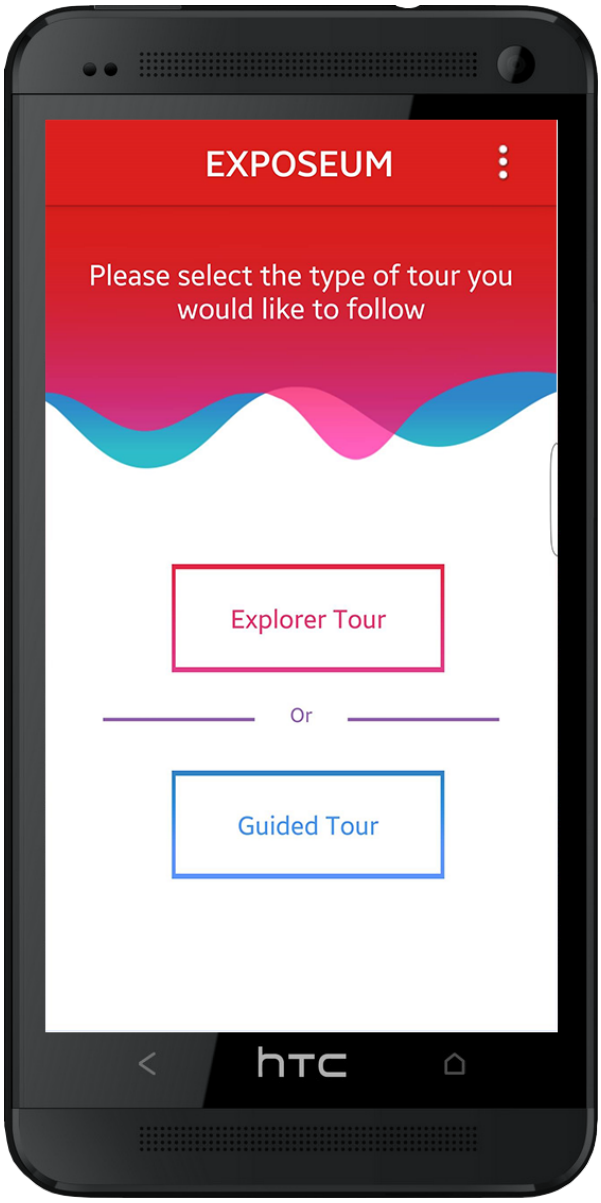
According to our stakeholder, the museum's most common demographics are students that visit the museum as part of school trips and elders. The following personas are precise descriptions of typical users of the product:

Stephanie is a 10 year old student. During the school year, she and her classmates have the opportunity to go on field trips to nearby museums. Due to the volume of students who enter the museum, they are split into groups. Exposeum allows Stephanie's group to guide themselves through the museum on a storyline designed for their age group, thus eliminating the need for a human guide for each student group. Stephanie also appreciates that she can complete a short quiz at the end of the storyline; she finds these quizzes fun and likes to compete with her friends.

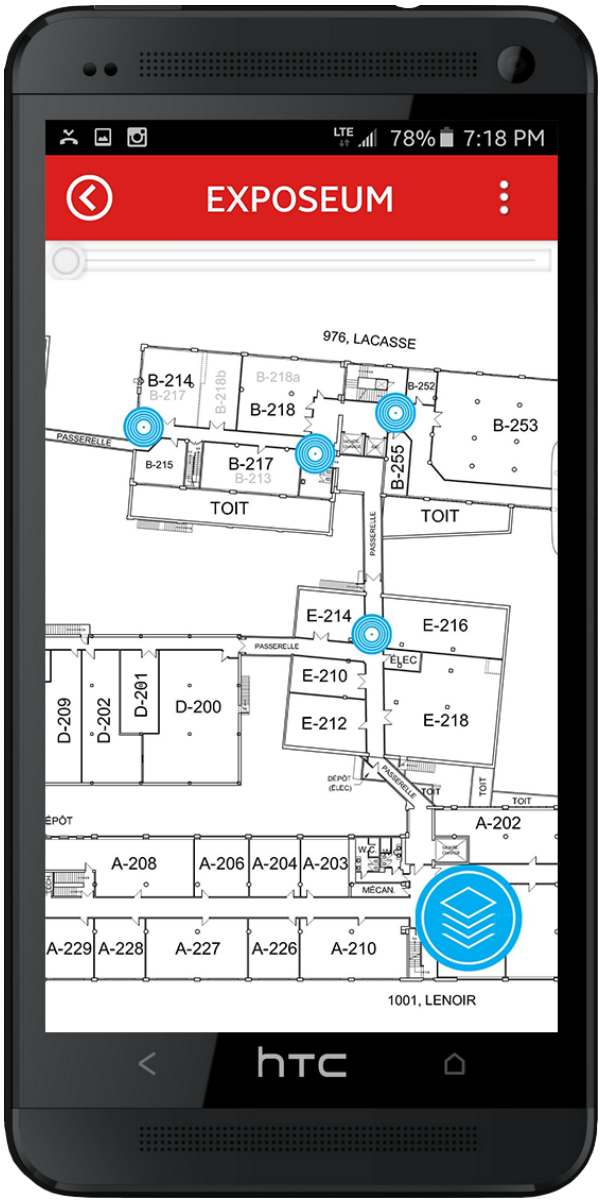
Pierre is a retired audio engineer originally from Lyon, France. He considers himself a history buff and loves exploring the origins of modern day technology. To occupy his free time, Pierre enjoys visiting museums but doesn't like following a tour because he is already knowledgeable in the domain. Exposeum offers him a free exploration mode whereby Pierre can freely explore and locate only the sites and exhibits that interest him. He also finds it exciting that Exposeum notifies him when he stumbles upon an exhibit he might have otherwise missed. Because his english is not very strong, he appreciates that he is able to change Exposeum's interface to his native tongue of French.

	
7.1 : As a User I would like to get an intro tutorial about this app	7.1 : As a User I would like to get a intro tutorial about this app

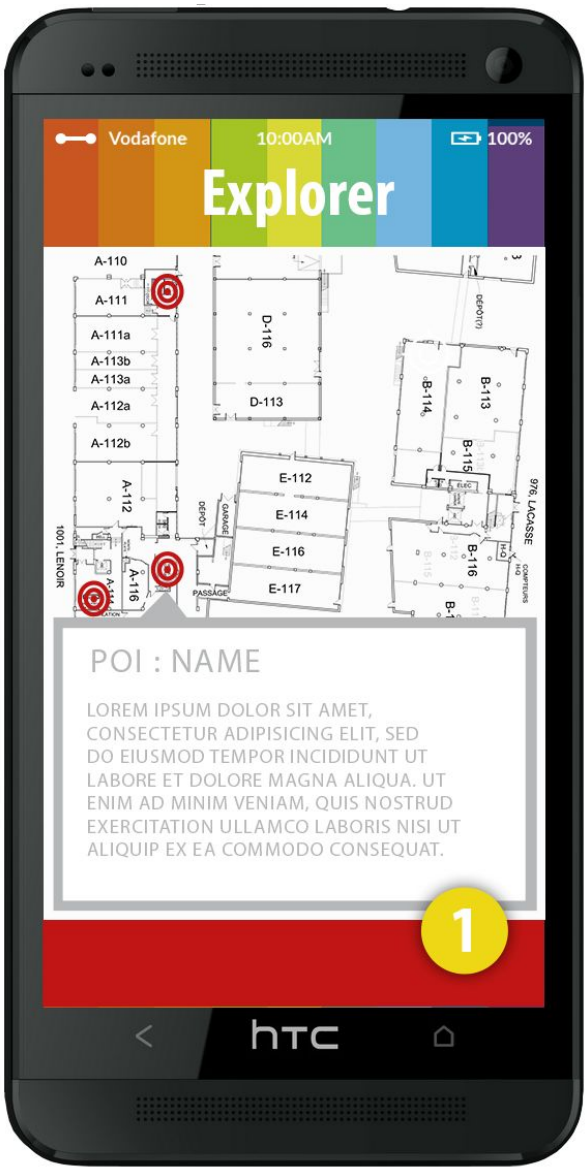
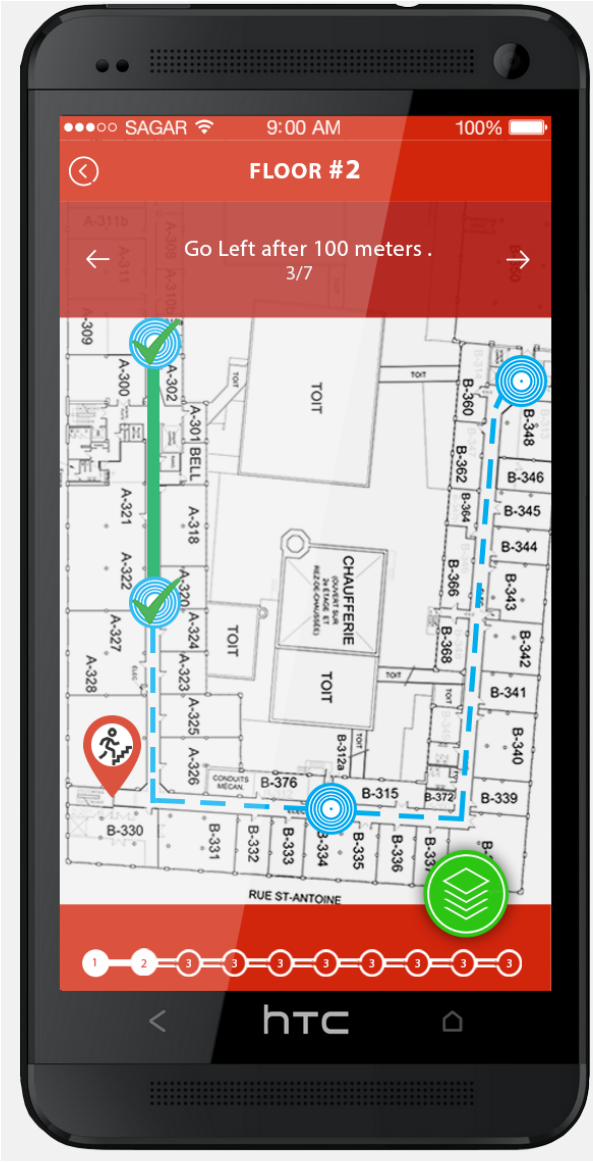
	
7.1 : As a User I would like to get a intro tutorial about this app	7.1 : As a User I would like to get a intro tutorial about this app

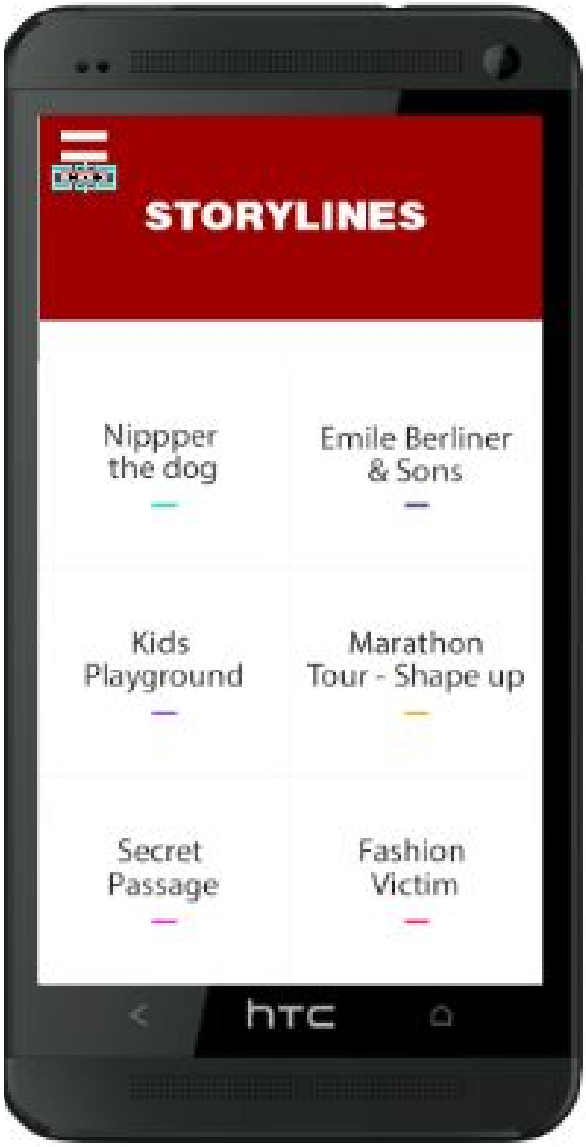


7.2 : As a user, I want to choose between a guided and an explorer tour.



7.3 : As a user I want to see all of the POIs

	
7.4 : As a user I want to select a POI and get info. (Explorer Mode)	7.5 : As a User I want to see the guided tour

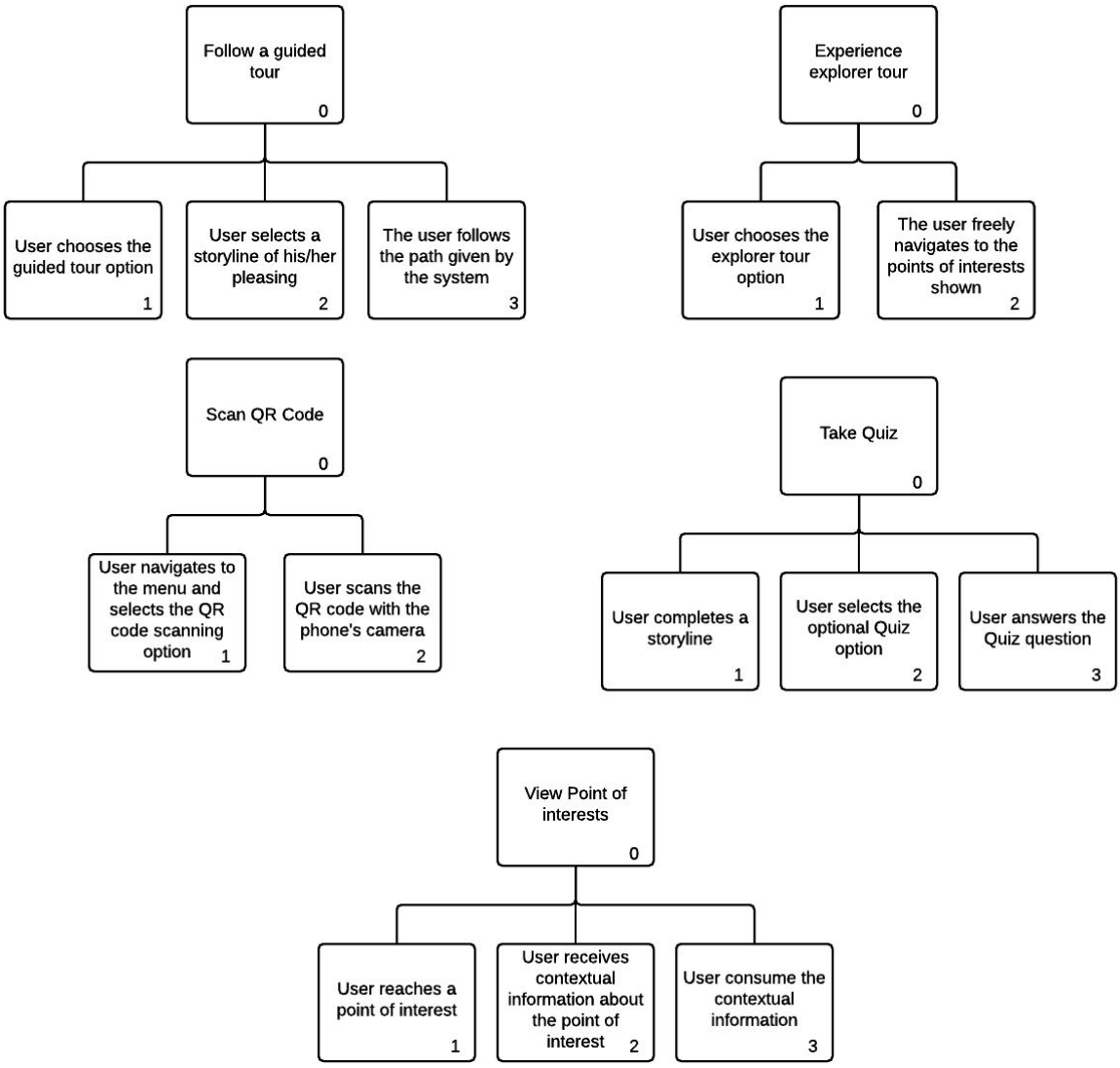
	
7.6 : As a user I would like to get Push notification.	7.7: As a user, I can see the list of storylines.

7.8 : As a user, I want to switch floors.

7.9: As a user, I can scan a QR code.

Hierarchical Task Diagram

The tasks the user can perform with the system are represented by the following hierarchical task model diagrams:



Scenarios

These tasks can also be reflected in scenarios, which describe tasks through informal narrative, while also capturing the context of the tasks.

Scenario 1:

Pierre is a retired audio engineer from France who was visiting his family in Montreal. He considers himself a history buff and loves exploring the origins of modern day technology so he thought that today is the day to visit the Musee des Ondes. He arrived at the building, and not wanting to have a guided tour, he decided to explore the building on his own. He loads the first floor plan in Exposeum and looks at all the different points of interest displayed on the screen. He starts heading to each one of them, and wherever he finds a QR code, he scans it and is given more information and pictures about the location. Some points of interest didn't have any QR codes to scan, but a page would open while he was in the beacon's vicinity to give him contextual information.

Scenario 2:

Mrs. Robinson is a history teacher that loves field trips. She has decided to take her class to visit the Musee des Ondes, however she understands that most of her students get bored pretty easily and will not be focused all along the visit. Even though she's not a big technology fan and she normally doesn't allow her students to use their cellphones, she allows them to use Exposeum so they can follow one of the different storylines targeted to kids. As soon as the visit starts, all the kids become engaged, listening to the noises of people walking, Nipper the dog, the main actor in that storyline, barking, and some very fun and interesting information showing at different stages of the visit. Near the end, each students was presented by Exposeum with a quick pop quiz which allows Mrs Robinson to know what the students had learned from their visit.

Scenario 3:

Mr. and Ms. Smith come from out of town to visit their son who happens to be a student at a Montreal University. They decide to pass the time by visiting the Musee des Ondes while their son is in class. Once they arrive to the museum, they open the Exposeum app, and pick one of the storylines that fit with their schedule and is for an adult audience. The storyline they choose guides them through different points of interest, providing them with its history put into a storytelling context. As promised, within less than 50 minutes, the storyline guided tour is over, giving them enough time to go back and meet their son.

7. Testing Plan and Report

Unit Testing

Unit testing tools

Unit testing was performed using NUnit. NUnit is a lightweight unit testing framework which comes bundled with Xamarin.

A separate project called Exposeum.Tests was added to the solution in order to isolate unit testing code from the core application code.

We currently observe the following convention for the unit tests found under this new testing project: Folder names follow the same name given in the Exposeum project, and some particular user stories have their own folder.

Relevant units to be tested

All core modules and their constituent units will have at a minimum one unit test. The map module will be tested to ensure that the app responds correctly to the user's interactions and that different map elements are clickable. The interaction between the application and the various iBeacons setup on-site will also be tested. The user's language preference and the persistence thereof will too be verified. Models database retrieval and persistence will be ensured, including CRUD operations and object serialization and deserialization.

Code Coverage Consideration

Unit test coverage for our application proved to be extremely difficult, we believe impossible, to accomplish. This is due to the fact that Exposeum is a mobile application written in C# using Mono, a framework which is compatible with but different from a pure .NET Microsoft approach.

Firstly, Xamarin UITests and the Xamarin-sanctioned NUnit unit tests do not offer the functionality of coverage percentage. Due to this, a third-party approach was attempted.

After much research, we attempted a number of Mono C# unit test coverage solutions, which all failed. First, we tried XR.Baboon, a mono C# unit test coverage tool available here. The tool worked, but only on the included sample C# project. The reason for this, we learned after much work, was that the tool requires a so-called ‘entry point’, ie. a ‘main’ method for access the the application for which the unit tests are designed. Because we are creating an Android application which must run either in an emulator or on a physical device, there is no way to connect this coverage tool with our Exposeum application.

We then tried other tools, namely monocov, NDepend, and DotCover. These all proved to fail for the same fundamental reason: in order to determine the source lines of code covered by our unit tests, the coverage tool needs to connect with the application itself. However, the application must run in an Android environment for which no connection is available from the tool. We require a unit testing framework which is designed for Android with Mono C#.

As a substitution, we have created the following table, which reflects the percentage of methods covered per indicated class.

Classes	Number of methods in Target class	Number of methods covered by unit test
AudioMapper	8	7
BeaconMapper	6	6
ExhibitionContentMapper	8	8
FloorMapper	8	8
ImageContentMapper	8	8
MapEdgeMapper	8	7
MapElementsMapper	7	6
PointOfInterestDescriptionMapper	8	8
PointOfInterestMapper	6	6
StatusMapper	3	3
StorylineDescriptionMapper	8	8
StorylineMapper	10	10
TextContentMapper	8	7
UserMapper	6	6
VideoMapper	8	8
WayPointMapper	8	8
BeaconTDG	5	4
ExhibitionContentEnTDG	6	5
ExhibitionContentFrTDG	6	5
FloorTDG	6	4
MapEdgeTDG	7	5
MapElementsTDG	6	5
MapTDG	5	0
PoiDescriptionEnTDG	5	4

PoiDescriptionFrTDG	5	4
StoryLineDescriptionEnTDG	5	4
StoryLineDescriptionFrTDG	5	4
StoryLineMapElementListTDG	6	5
StorylineTDG	6	4
UserTDG	6	4
BeaconFinderTests	29	14
ShortestPathProvider	3	3
StorylineController	10	4
QRController	3	1
Total	**242**	**193**

Acceptance Testing

When all the tasks associated with a particular story are completed, a corresponding acceptance test outlining the acceptance criteria is demonstrated to the stakeholder and signed off on.

Sprint 1

AT-1	**US-1 - As a Visitor, I want to specify my preferred language (english or french) at any time.**
Acceptance Criteria	Given that I am in the application, the user interface matches the device's language. If I modify the device's language, then the interface of the application matches the new device language.
Result	**PASS**
Comments	

AT-2	**US-13 - As a Visitor, I want to receive full contextual information about a point of interest in my proximity.**
Acceptance Criteria	Given that I am in the application, and then I enter the proximity of a beacon (within 0.5m), the interface displays contextual information about the associated POI.
Result	**PASS**
Comments	

AT-3	**US-14 - As a Visitor, I want to view the entire map of every floor with all points of interest when in free visit mode.**
Acceptance Criteria	Given that I am in the opened the application, and that I have selected free visit mode, then I can view the points of interest in the map, and I can change floors in the map.
Result	**PASS**
Comments	

Sprint 2

AT-4	US-6 - As a Visitor, I can engage a free tour mode of the building.
Acceptance Criteria	Given that I am in the application, then I can select Explorer mode to be able to go through the museum in freeroam mode.
Result	**PASS**
Comments	

AT-5	US-7 - As a visitor, I want to select any point of interest and view its summary when in free visit mode.
Acceptance Criteria	Given that I am in the application, and I have selected the Explorer mode, then I can select a point of interest. This will trigger a popup that will display a summary of the selected point of interest.
Result	**PASS**
Comments	

AT-6	US-9 - As a Visitor, I want to receive push notifications when the app is not in focus.
Acceptance Criteria	Given that I am not in the application, then the app will send me a push notification once I come in contact with a beacon.
Result	**PASS**
Comments	

AT-7	US-11 - As a Visitor, I want to see which points of interest I have already visited.
Acceptance Criteria	Given that I am in the application, the points of interest that I already visited will be indicated with an updated icon (adorned with a checkmark).
Result	**PASS**
Comments	

AT-8	US-19 - As a user, I want to receive an intro to the app.
Acceptance Criteria	Given that I am in the application, and I have selected the language of my preference, the application will display a “Splash page” to give me indications of how to use the app.
Result	**PASS**
Comments	

AT-9	US-20 - As a user I want to choose between Guided tour and an Explorer tour.
Acceptance Criteria	Given that I am in the opened the application, I am able to select between Guided and Explorer tour.
Result	**PASS**
Comments	

Sprint 3

AT-10	US-2 - As a visitor, I want to view a list of up to date storylines available.
Acceptance Criteria	Given that I am in the application, and I have selected the Guided mode, then the app displays a list of the up to date storylines available.
Result	**PASS**
Comments	

AT-11	US-3 - As a visitor, I want to preview a storyline before starting it.
Acceptance Criteria	After selecting the Guided mode, I can choose any of the presented storylines. A popup with a preview of the chosen storyline is displayed before I start it.
Result	**PASS**
Comments	

AT-12	US-5 - As a visitor, I want to follow guided tours (storylines).
Acceptance Criteria	After starting a storyline, a map view is displayed with paths between the points of interests, which is updated dynamically.
Result	**PASS**
Comments	

AT-13	US-8 - As a visitor, I want to stop a storyline in progress and start a new one.
Acceptance Criteria	Given that I am in a storyline, I can go back to the storyline list and select a new storyline. Newly selected storyline will be displayed in the map.
Result	**PASS**
Comments	

AT-14	US-10 - As a visitor, I want to pause a storyline in progress and resume at a later time.
Acceptance Criteria	Given that I am in a storyline, pausing a storyline will temporarily will stop beacon notifications from being shown to the user. Resuming the storyline reenables notifications.
Result	**PASS**
Comments	

AT-15	US-12 - As a visitor, I want to have a progress bar for a storyline.
Acceptance Criteria	Given that I am in a storyline, my progression in the storyline visually conveyed both in terms of path progression per floor and overall completion in terms of points visited.
Result	**PASS**
Comments	

Sprint 4

AT-16	US-16 - As a visitor, I want to be able to scan QR codes.
Acceptance Criteria	Given that I am in a application, opening the top menu displays a QR scan option. Selecting it will open the camera, allowing the user to scan the codes.
Result	**PASS**
Comments	

AT-17	US-18 - As a visitor, once I have finished a storyline, I want to be directed to the closest exit or back to the venue start point.
Acceptance Criteria	Given that I have completed a storyline, and agreed to be guided to an exit, the application will display a path to an appropriate exit.
Result	**PASS**
Comments	

AT-18	US-22 - As a visitor, I want to be able to reset the application to a fresh state.
Acceptance Criteria	Given that I am in the main page of the application (tour selection), opening the top menu displays a Reset All option. Selecting it will erase all progress up to date, resetting the app to a fresh state.
Result	**PASS**
Comments	

Sprint 5

AT-19	US-24 - As a user I want to be able to skip a POI and continue the current Guided Tour.
Acceptance Criteria	Given that I am in a storyline, if a POI is skipped, a popup displays asking whether I want to skip this point of interest or return to the correct (in sequence) POI.
Result	**TO BE TESTED**
Comments	

AT-20	US-25 - As a visitor, when resuming a Guided Tour, I want to be directed from closest POI to the last visited POI.
Acceptance Criteria	Given that I resume a storyline, a popup asks whether I want to be redirected to the last visited POI.
Result	**TO BE TESTED**
Comments	

System Tests

Because no amount of unit testing can replace the need to perform a system interaction test, such as simulating a real user interacting with the app directly, further testing is needed in the form of system testing.

In our application, for example, this would be testing that pinching over the map results in the zoom level being increased or decreased, for instance. For this, Xamarin offers a solution called Xamarin.UITest which allows for the programmatic simulation of user interaction events directly with the interface of the application.

For this, a new project was added to the solution called Exposeum.UITests. There, a sequence of user interactions on specific UI elements can be specified, and the prescribed outcome can be asserted. Using this project, entire user stories can be tested in one shot.

In fact, Xamarin.UITests can model most of our system tests in a 1-to-1 manner, because user interaction events can be sequenced programmatically, replacing the need for a human user.

Sprint 1 System Testing

ST-1	**US-1 - As a Visitor, I want to specify my preferred language (english or french) at any time.**		**Expected Output**	**Result**
Steps to reproduce	1	Open the application	The language of the application matches the language of the phone	Pass
	2	Navigate phone system settings and switch language	-	Pass
	3	Go back to application	The language of the application matches the new selected language	Pass
Result				**Pass**
Comments				

ST-2	**US-13 - As a Visitor, I want to receive full contextual information about a point of interest in my proximity.**		**Expected Output**	**Result**
Steps to reproduce	1	Open the beacon activity	Beacon activity is displayed	Pass
	2	Get within 0.5 meters of a beacon	Information associated with the beacon is retrieved from the database and displayed	Pass
Result				**Pass**
Comments				

ST-3	**US-14 - As a Visitor, I want to view the entire map of every floor with all points of interest when in free visit mode.**		**Expected Output**	**Result**
Steps to reproduce	1	Open the map activity	Map is displayed at the 1st floor with corresponding POIs	Pass
	2	Select the ‘floor up’ button	Second floor map is displayed with corresponding POIs	Pass
Result				**Pass**
Comments				

Sprint 2 System Testing

ST-4	**US-6 - As a Visitor, I can engage in a free tour mode of the building so that I can visit all POIs in an unrestricted way.**		**Expected Output**	**Result**
Steps to reproduce	1	Select Explore Mode (Open the map activity)	Map is displayed at the 1st floor, with its corresponding POIs	Pass
	2	Select any floor	Selected floor, with its corresponding POIs, is displayed	Pass
Result				**Pass**
Comments				

ST-5	**US-7 - As a visitor, I want to select any point of interest and view its summary when in free visit mode.**		**Expected Output**	**Result**
Steps to reproduce	1	Open the map activity	Map is displayed at the 1st floor, with its corresponding POIs	Pass
	2	Select a given POI	The summary of the selected POI is displayed	Pass
Result				**Pass**
Comments				

ST-6	US-9 - As a Visitor, I want to receive push notifications when the app is not in focus.		Expected Output	Result
Steps to reproduce	1	Start the application	Application opens to the main view	Pass
	2	Lock the device	Phoned Locked	Pass
	3	Get within 0.5m of a beacon	Application sends push notification to the user	Pass
Result				Pass
Comments				

ST-7	US-11 - As a Visitor, I want to see which points of interest I have already visited.		Expected Output	Result
Steps to reproduce	1	Start the application	Application opens to the main view	Pass
	2	Get within 0.5m of a beacon	POI is displayed as visited	Pass
Result				Pass
Comments				

ST-8	US-19 - As a user, I want to receive an intro to the app.		Expected Output	Result
Steps to reproduce	1	Start the application	Application opens to the main view	Pass
	2	Select language	The selected language is used and saved	Pass
Result	Receive an intro to the app			Pass
Comments				

ST-9	**US-20 - As a user I want to choose between Guided tour and an Explorer tour.**		**Expected Output**	**Result**
Steps to reproduce	1	Start the application	Application opens to the main view	Pass
	2	Select language	The selected language is used	Pass
	3	Receive an intro to the app	An intro is displayed	Pass
	4	I go through the splash page	I am presented with the options to select Guided tour/select Explorer tour	Pass
Result				**Pass**
Comments				

Sprint 3 System Testing

ST-10	**US-2 -** As a visitor, I want to view a list of up to date storylines available.		**Expected Output**	**Result**
Steps to reproduce	1	Select guided tour	Updated list of storylines is presented	Pass
Result				**Pass**
Comments				

ST-11	**US-3 -** As a visitor, I want to preview a storyline before starting it.		**Expected Output**	**Result**
Steps to reproduce	1	Select guided tour	Storyline list is displayed	Pass
	2	Select a specific storyline	Preview of storyline is displayed	Pass
Result				**Pass**
Comments				

ST-12	**US-5 -** As a visitor, I want to follow guided tours (storylines).		**Expected Output**	**Result**
Steps to reproduce	1	Select guided tour	Storyline list is displayed	Pass
	2	Select and start a storyline	Map view updates and displays directions on the map	Pass
Result				**Pass**
Comments				

ST-13	**US-8 -** As a visitor, I want to stop a storyline in progress and start a new one.		**Expected Output**	**Result**
Steps to reproduce	1	Select guided tour	Storyline list is displayed	Pass
	2	Select and start a storyline	Map view updates and displays directions on the map	Pass
	3	Activate the back control	Storyline page is redisplayed	Pass
	4	Select any available storyline	Newly selected storyline is loaded in the place of the old one	Pass
Result				**Pass**
Comments				

ST-14	**US-10 -** As a visitor, I want to pause a storyline in progress and resume at a later time.		**Expected Output**	**Result**
Steps to reproduce	1	Select guided tour	Storyline list is displayed	Pass
	2	Select and start a storyline	Newly selected storyline is loaded in the place of the old one	Pass
	3	Activate the back control	Storyline page is redisplayed	Pass
	4	Reselect the storyline that was previously selected	Previously selected storyline is resumed at its previous progression	Pass
Result				**Pass**
Comments				

ST-15	**US-12 -** As a visitor, I want to have a progress bar for a storyline.		**Expected Output**	**Result**
Steps to reproduce	1	Select guided tour	Storyline list is displayed	Pass
	2	Select and start a storyline	Storyline is loaded in the map, and a progress bar is shown at the bottom	Pass
Result				**Pass**
Comments				

Sprint 4 System Testing

ST-16	**US-16 -** As a visitor, I want be able to scan QR codes.		**Expected Output**	**Result**
Steps to reproduce	1	Open top menu	Side drawer is opened up and displayed	Pass
	2	Select QR scan option	QR scan page is displayed (camera is loaded)	Pass
	3	Position phone directly in front of QR code	QR code is detected and deciphered. Associated content is loaded.	Pass
Result				**Pass**
Comments				

ST-17	**US-18 -** As a visitor, once I have finished a storyline, I want to be directed to the closest exits or back to the venue start point.		**Expected Output**	**Result**
Steps to reproduce	1	Select guided tour	Storyline list is displayed	Pass
	2	Select and start a storyline	Storyline is loaded in the map, and on top an indication bar is displayed. This section includes hints about how to get to next POI.	Pass
Result				**Pass**
Comments				

ST-18	**US-22 -**As a visitor, I want to be able to reset the application to a fresh state.		**Expected Output**	**Result**
Steps to reproduce	1	Open the tour selection page	I am presented with the options to select Guided tour/select Explorer tour, and a top menu	Pass
	2	Open top menu	Side drawer is opened up and displayed	Pass
	3	Select reset option	All progress is set to default. All points of interest are set to non-visited.	Pass
Result				**Pass**
Comments				

Sprint 5 System Testing

ST-19	**US-24 -** As a user I want to be able to skip a POI and continue the current Guided Tour		**Expected Output**	**Result**
Steps to reproduce	1	Select and start a storyline	Storyline is loaded in the map, and on top an indication bar is displayed.	Pass
	2	User skips a know point of interest	Popup indicates that the user is out of order, with option to continue or return	Pass
	3.1	If intentional, user selects continue option	Skipped point of interest are marked as visited, and storyline proceeds as normal	Pass
	3.2 (Alternate)	If unintentional, user selects return option	User is directed back to last unvisited POI (using shortest path functionality).	Pass
Result				**Pass**
Comments				

ST-20	**US-25 -** As a visitor, when resuming a Guided Tour, I want to be directed from closest POI to the last visited POI.		**Expected Output**	**Result**
Steps to reproduce	1	Resume a storyline in progress	Storyline is loaded in the map,with previous progress preserved and displayed	Pass
	2	Users comes into contact with any point of interest (beacon)	System guides user back to the last unvisited point of interest. Storyline proceeds as normal.	Pass
Result				**Pass**
Comments				

8. Defect Tracking and Report

Sprint 1 report:

In this sprint two out of three reported bugs were resolved. EX-53 was addressed but not reviewed, hence it was not marked as resolved.

Defect ID	Description	Discovered	Resolved	Status
EX-51	Wrong beaconid type	Sprint 1	Sprint 1	**RESOLVED**
EX-52	Name of POI_insertion.cs POI_ListDisplay.cs does not follow the naming convention	Sprint 1	Sprint 1	**RESOLVED**
EX-53	Improper Database Structure	Sprint 1	-	**IN PROGRESS**

Sprint 2 report:

In this sprint two out of four reported bugs were resolved, one of this coming from Sprint 1.
Both EX-53 and Ex -83 are still pending and will be addressed in Sprint 3.

Defect ID	Description	Discovered	Resolved	Status
EX-53	Improper Database Structure	Sprint 1	-	**IN PROGRESS**
EX-79	Floating 3-dot settings circle needs to be removed	Sprint 2	-	**TO DO**
EX-81	No beacon detected when app is paused (in the background)	Sprint 2	Sprint 2	**RESOLVED**
EX-82	Storyline beacon is coupled to EstimoteSdk.Beacon	Sprint 2	-	**IN PROGRESS**

EX-83	User not redirected to the proper activity after tapping a notification	Sprint 2	Sprint 2	**RESOLVED**

Sprint 3 report:

In this sprint six out of eleven reported bugs were resolved. Both EX-53 and EX-82 are still pending from previous sprints and were not addressed in this sprint. EX-82's priority was lowered as it appears to be trivial and have no direct impact on the application. EX-53, EX-116 and EX-118 are scheduled to addressed in Sprint 4.

Defect ID	Description	Discovered	Resolved	Status
EX-53	Improper Database Structure	Sprint 1	-	**IN PROGRESS**
EX-79	Floating 3-dot settings circle needs to be removed	Sprint 2	-	**TO DO**
EX-82	Storyline beacon is coupled to EstimoteSdk.Beacon	Sprint 2	-	**IN PROGRESS**
EX-84	Point of interest icon does not change accordingly when Beacon is found.	Sprint 3	Sprint 3	**RESOLVED**
EX-93	Text information from beacon does not always appear in popup	Sprint 3	Sprint 3	**RESOLVED**
EX-111	Run-time exception (list of POIs is null)	Sprint 3	Sprint 3	**RESOLVED**
EX-114	Rename the classes with underscore to follow proper naming convention	Sprint 3	Sprint 3	**RESOLVED**

EX-115	When tapping a notification, the user is not being redirected to the proper activity	Sprint 3	Sprint 3	**RESOLVED**
EX-116	Rotating the phone while on the MapView crashes the app.	Sprint 3	-	**TO DO**
EX-117	OutOfOrderPoint window repeatedly displays after dismissing	Sprint 3	Sprint 3	**RESOLVED**
EX-119	Dismiss Button on the OutOfOrderPointOfInterestPopup does not dismiss the Popup	Sprint 3	Sprint 3	**RESOLVED**

Sprint 4 report:

In this sprint all of the open bug reports were resolved. No new bug report were reported.

Defect ID	Description	Discovered	Resolved	Status
EX-53	Improper Database Structure	Sprint 1	Sprint 4	**RESOLVED**
EX-79	Floating 3-dot settings circle needs to be removed	Sprint 2	Sprint 4	**RESOLVED**
EX-82	Storyline beacon is coupled to EstimoteSdk.Beacon	Sprint 2	Sprint 4	**RESOLVED**
EX-116	Rotating the phone while on the MapView crashes the app.	Sprint 3	Sprint 4	**RESOLVED**
EX-143	Unable to properly reset a storyline	Sprint 4	Sprint 4	**RESOLVED**

Sprint 5 report:

In this sprint no new bugs were reported.

9. Quality Metrics

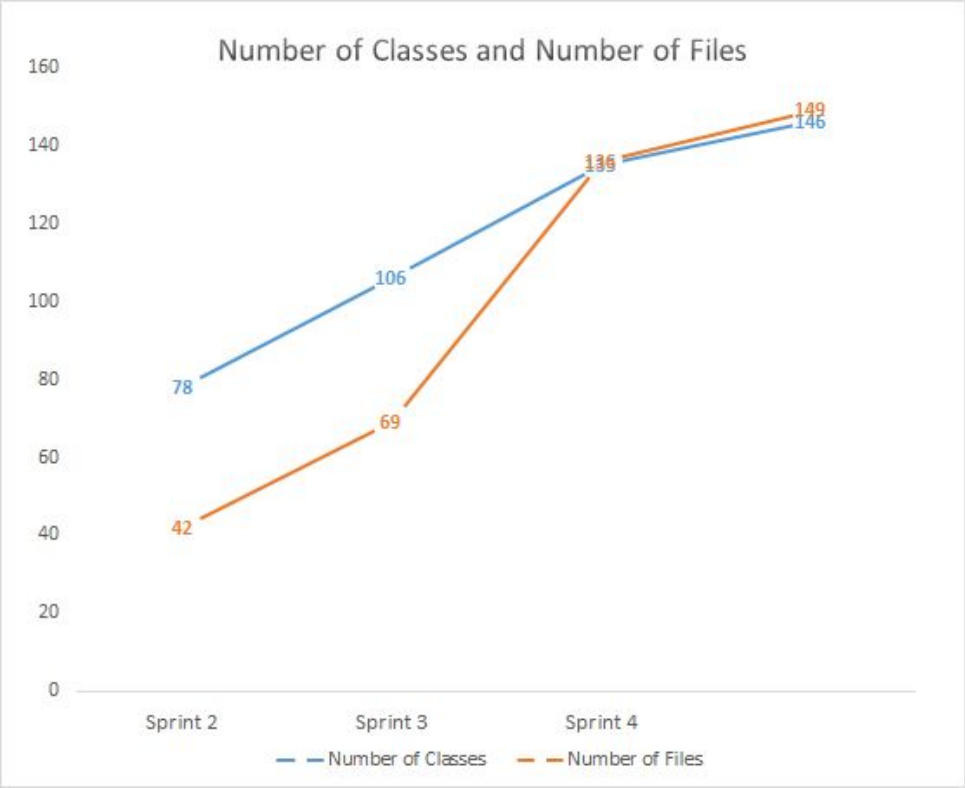
SonarQube is a code quality inspection tool which we use to collect various metrics, in a bid to keep track of the quality of our code base throughout each sprint. The following metrics, in particular, are considered.

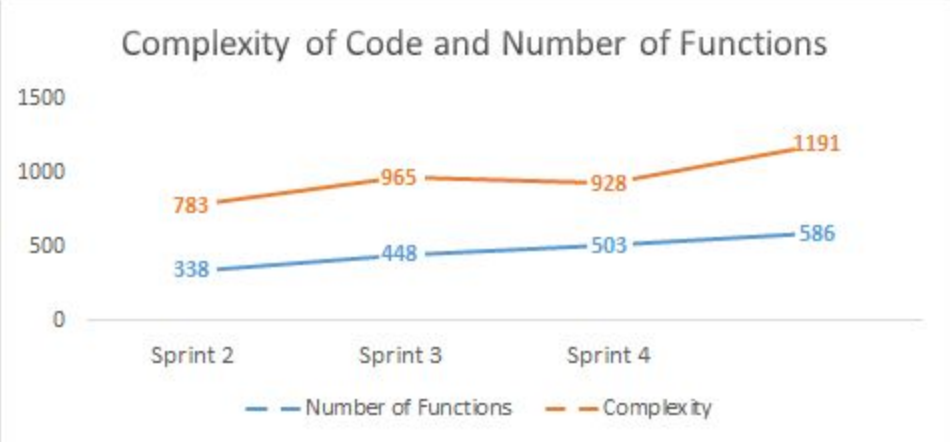
Metric	Definition
Lines of Code	Represents the number of physical lines of source code, excluding blank lines and comments.
Number of Functions	Simply reflects the number of methods contained in our source code. Our number reflects the fact that accessors, mutators, and constructors are counted as methods.
Number of Classes	Represents the total number of classes.
Number of Statements	Represents the number of statements encountered, for example break, for, while, do, and try fall under this category.
Number of Files	Represents the total number of files which contain the source code of our system.
Duplication	The density of duplicated lines of code in the project. Duplicate lines consist of lines that are 100% identical.
Documentation	Percentage of comment lines in the project.
Technical Debt (Time)	The time it would take to address issues / faults that SonarQube has identified.
Technical Debt (Issues)	The actual amount of issues / faults SonarQube has identified.
Complexity	The cyclomatic complexity of the program (# of linearly independent paths).

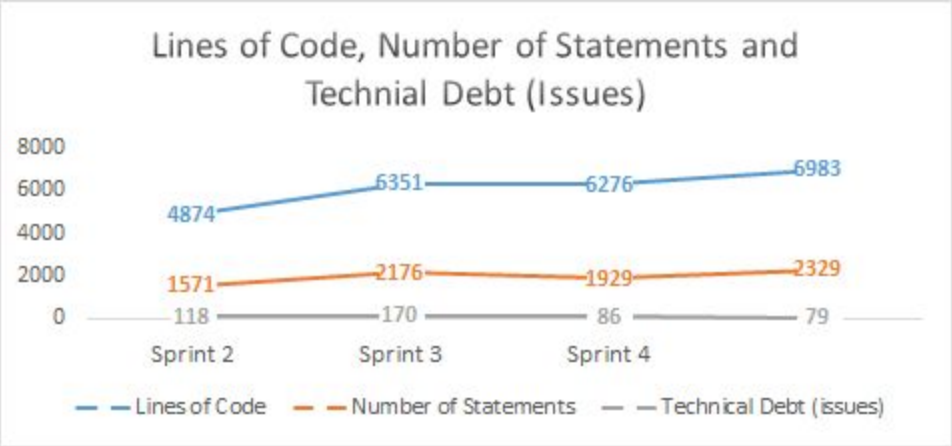
Code Metrics per Sprint

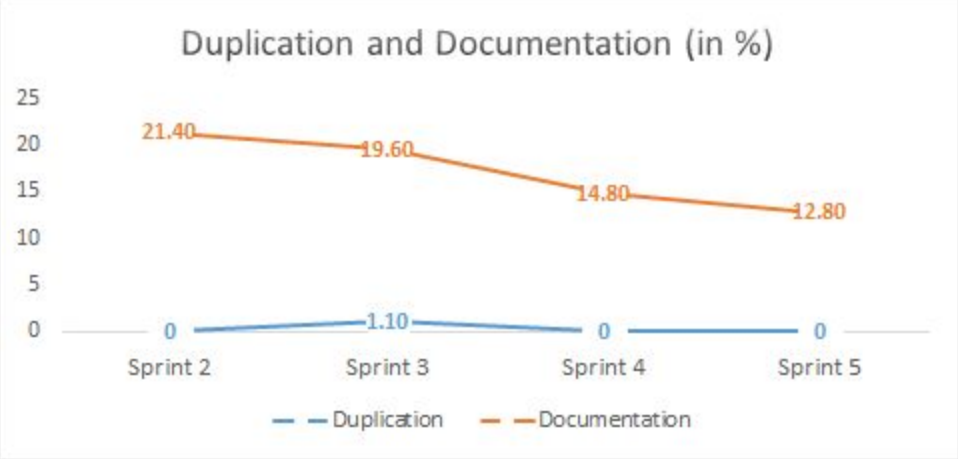
Metric	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5
Lines of Code	N/A	4874	6351	6276	6983
Number of Functions	N/A	338	448	503	586
Number of Classes	N/A	78	106	135	146
Number of Statements	N/A	1571	2176	1929	2329
Number of Files	N/A	42	69	136	149
Duplication	N/A	0	1.1%	0	0
Documentation	N/A	21.4%	19.6%	14.8%	12.8%
Technical Debt (time)	N/A	1d2h	1d6h	1d	22h
Technical Debt (issues)	N/A	118	170	86	79
Complexity	N/A	783	965	928	1191

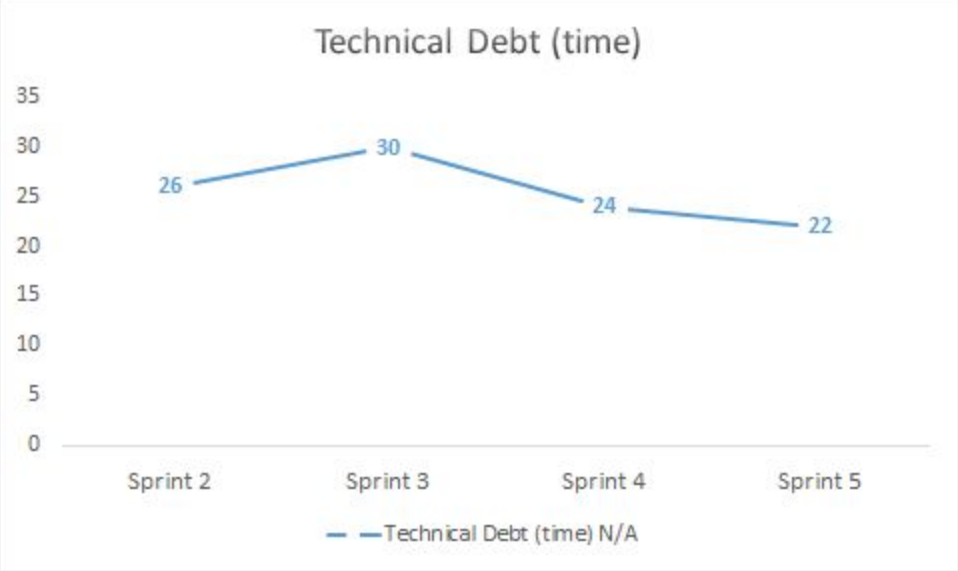
Code Metrics Graph






Lines of Code, Number of Statements and Technial Debt (Issues)
8000
6000
4000
2000
0
4874
6351
6276
6983
1571
2176
1929
2329
118
170
86
79
Sprint 2
Sprint 3
Sprint 4
Lines of Code
Number of Statements
Technical Debt (issues)


Duplication and Documentation (in %)
25
20
15
10
5
0
21.40
19.60
14.80
12.80
0
1.10
0
0
Sprint 2
Sprint 3
Sprint 4
Sprint 5
Duplication
Documentation


Technical Debt (time)
35
30
25
20
15
10
5
0
26
30
24
22
Sprint 2
Sprint 3
Sprint 4
Sprint 5
Technical Debt (time) N/A

Code Metrics Discussion

Lines of Code, Number of Functions, Classes, Statements, and Files

In general, a software system that is smaller in terms of number of lines of code, functions, classes, statements, and files is more desirable as it is more manageable and maintainable. However, it is expected and normal that these metrics increase in tandem with feature implementation as development progresses.

From sprint 4 to sprint 5, number of lines of code dropped from 6276 to 6983. During this same sprint period, number of functions increased by 503 to 586, number of classes increased from 135 to 146, and number of statements decreased to 2329.

Sprint 5 saw a major overhaul to the database relational architecture (TDGs, JSON parsing, etc.), thereby we had an increase in all these metrics. However, as you'll see further down, quality remained unaffected.

Duplication

From sprint 4 to sprint 5 the duplication remained at 0%. Because of changes made in sprint 4, having duplicate code for features such as language change were made unnecessary. This change persisted throughout sprint 5 and therefore no code duplication was needed.

Documentation

From sprint 4 to sprint 5, the value for this metric decreased from 14.8% to 12.8%. Actual comments lines in this sprint is now 1245

In general, a lower value is better. High quality software should be readable, and should speak for itself in terms of human readability. Of course, some code segments may need a comment to explain a rationale or a particularly quirky section; since not all comments are dispensable, a value of 0% for this metric is not necessarily desired.

So far, we as a team continue to interact with each other's code to a significant degree and have not had any outstanding problems understanding each other's code, and so we are satisfied with our value for this metric as of yet.

Technical Debt (time/issues)

From sprint 4 to sprint 5, our technical debt, in SonarQube's view, decreased by 2 hours.

The vast majority of issues detected SonarQube were very simple refactorings fixed by ReSharper (VS extension). This is an automated process and did not require much effort. (example: unused references, declared but unused variables, etc.)

Complexity

From sprint 4 to sprint 5, complexity increased from 928 to 1191 (+263).

This increase in complexity is due to the finalization of data serialization, done with the use of data source architectural patterns such as data mappers and table data gateways. Despite JSON parsing also adding to the net complexity, as with previous sprints, complexity per class and per function decreased.

Appendix

Investigation: In-app billing

At the start of the previous sprint, the client asked about the possibility of adding in-app billing to the application. After some research we found that it would be possible in integrate in app purchases for premium content into Exposeum through Google's "In-app billing" service.

Overview

In-app purchases are handled by Google and are directly billed to the user's Google Play account. The app never handles sensitive payment details, Google Play takes care of the whole checkout process. Such a system allows for greater security and a simplified checkout process where the user does not need to enter any credit card information when making the purchase.

Google charges a 30% fee on any purchases made through this system[1]. Purchased items are also managed by Google. This means that Google Play will "keep track of the user's ownership of in-app products"[2]. A user will be able to restore previous purchases if the app is reinstalled.

In app-billing is integrated in an app using Android's Google Play API which comes bundled with the Android SDK.

Requirements

For in-app billing to work there are a few requirements that need to be met. First, it is necessary that the application be published on the Google Play Store since all payment requests will be handle by the same service. Second the device must be loaded with the Google Play Services app and the Google Play store. Most android phones come bundled with these apps but in rare cases (ex: custom ROMs) these apps are not included due to licensing restrictions. Third, the phone must be running Android 2.2 or higher. Finally, the phone must have access to an internet connection in order to make the purchase. [2]

Purchase Flow

The system diagram below depicts the high level flow for a purchase made through Google Play.

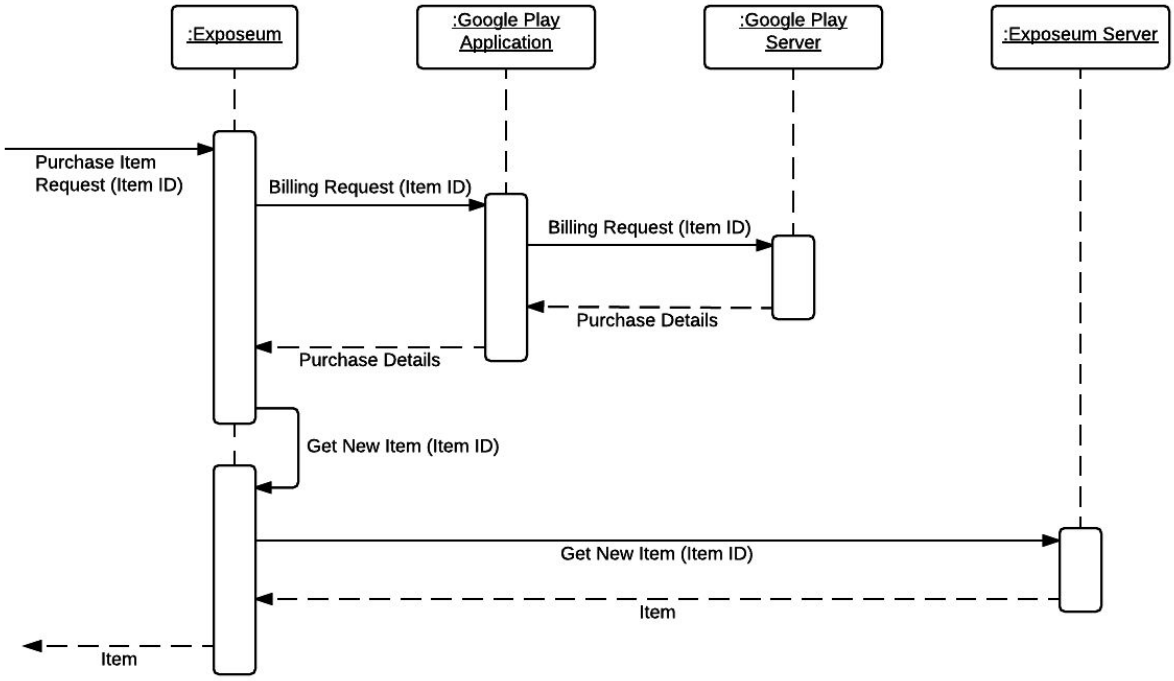


Figure 10 - In App Purchase Sequence Diagram

The Exposeum application never directly communicates with Google Play's Server. Instead, Billing Requests for an Item are sent through the Google Play Application located on the phone through an exposed API. The Google Play Application forwards this request to Google Play's Server which takes care of the checkout process. The server will return the purchase details to Exposeum through the Google Play Application. It is then the responsibility of Exposeum to unlock the purchased item.

In order to increase security and prevent premium items from being unlocked by decompiling the app, Sinister Six proposes that Exposeum retrieves premium items separately from the remote server.

Administration

In-app products are managed through the Google Play Developer Console. This console allows a publisher to add and maintain product lists, handle refunds and set up test accounts to test in-app billing.

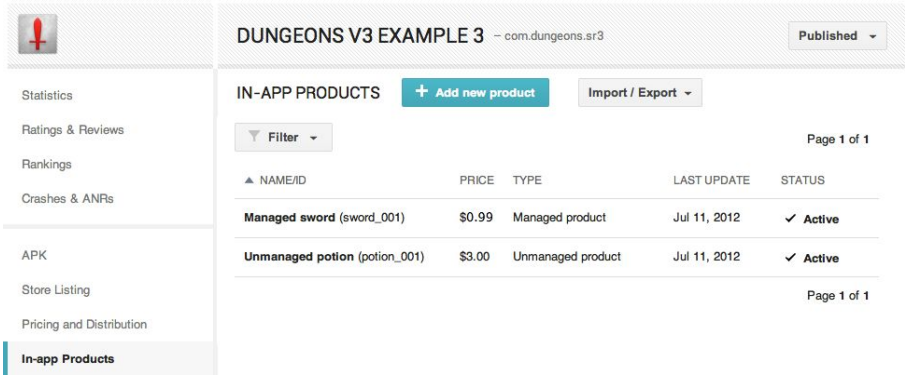


Figure 11 - In App Purchasing dashboard

Effort Estimation

The following USP estimations were elicited using the planning poker technique. We estimated that TASK3 would require a medium amount of work compared to the other tasks. TASK4 requires the most USPs because it will require modifying the Exposeum Application as well as coordinate with the Map Editor team to provide the new functionality.

Description	USP
As a User I want to purchase premium StoryLines	Total: 13
TASK1 - Add support for premium StoryLines in Exposeum	1
TASK2 - Add UI support for item purchase	2
TASK3 - Integrate Google Play's In-App billing	3
TASK4 - Modify the Exposeum server API to supply premium StoryLines	5
TASK5 - Integrate the new API in Exposeum	2

Alternatives

For apps distributed through the Google Play Store, it is against Google's Terms Of Services to allow purchase of digital goods through a third party.

"App purchases: Developers charging for apps and downloads from Google Play must do so by using Google Play's payment system. In-app purchases: Developers offering virtual goods or currencies within a game downloaded from Google Play must use Google Play's in-app billing service as the method of payment. ***Developers offering additional content, services or functionality within another category of app downloaded from Google Play must use Google Play's in-app billing service as the method of payment****, except: where payment is primarily for physical goods or services (e.g., buying movie tickets, or buying a publication where the price also includes a hard copy subscription); or where payment is for digital content or goods that may be consumed outside of the app itself (e.g., buying songs that can be played on other music players)."* [3]

It could be possible however to offer in person payment at the museum, and allow for the content to be redeemed using a voucher issued by the museum's staff. This could allow purchase of premium storylines for people that don't have a credit card added to their Google Play account and it would also allow payments using cash and Interac Debit.

Feasibility Assessment

It is possible for Sinister Six to integrate in-app purchases in Exposeum, however the inclusion of the feature would require the addition of at least one more user story to the backlog. Doing so would affect the scope of the project and in order to contain project completion within 5 sprints an existing user story (or stories) of equal value would need to be substituted out. After proposing the compromise to the client, it was agreed that other more important features be prioritized instead.

[1] https://support.google.com/googleplay/android-developer/answer/112622
[2] http://developer.android.com/google/play/billing/billing_overview.html
[3] https://play.google.com/about/monetization.html