Augmented Reality in Mobile

HLD & LLD Design Mobile app with Unity

Η εφαρμογή διατίθεται στο Google Play Store production ready



**SCHOOL OF ARCHITECTURE, COMPUTING & ENGINEERING**

*BSc in Computer Science*

**Michail Markou, Chaidos John, Manios Athanasios, Alantzopoulos**

*CN6008 – Advanced Topics in Computer Science*

**UEL NUMBER**

*2020732 (MM), 1966133 (CJ), 2020737 (MA)*

***Date***

2022/05/01

TABLE OF CONTENTS

[agenda 1](#_Toc103260740)

[I. Brief 1](#_Toc103260741)

[II. Description Scope 2](#_Toc103260742)

[III. Financial Scope 3](#_Toc103260743)

[IV. Focused on Delivery 3](#_Toc103260744)

[V. Project Execution 3](#_Toc103260745)

[VI. Technologic Approach 4](#_Toc103260746)

[1. DESIGN 5](#_Toc103260747)

[1.1. PROBLEM ANALYSIS 5](#_Toc103260748)

[Μεθοδολογία 6](#_Toc103260749)

[Game Engine Options & Tools 6](#_Toc103260750)

[1.2. PERSONA 6](#_Toc103260751)

[1.3. Requirement Analysis (non/functional) & Use Cases Ερμηνείες 7](#_Toc103260752)

[1.4. UML Representation Analysis 10](#_Toc103260753)

[1.4.1. Systems Use Case 10](#_Toc103260754)

[1.5. Class Diagrams & ERD 11](#_Toc103260755)

[1.6. Storyboards & Wireframes (low fidelity) 13](#_Toc103260756)

[1.7. User Flows 17](#_Toc103260757)

[1.8. State Machine/State Chart Diagram 18](#_Toc103260758)

[1.9. Testing 18](#_Toc103260759)

[2. Project Management 19](#_Toc103260760)

[2.1. Lifecycle 19](#_Toc103260761)

[2.2. Task/Time Schedule PERT 21](#_Toc103260762)

[2.3. Work Breakdown Structure (WBS) 21](#_Toc103260763)

[2.4. Sprint Canvas 22](#_Toc103260764)

[2.5. Scrum Cabinet Reports 23](#_Toc103260765)

[2.6. Project Participants – κατανομή εργασιών 24](#_Toc103260766)

[2.6.1. Stakeholder List 24](#_Toc103260767)

[2.6.2. Product Owner 25](#_Toc103260768)

[2.6.3. Scrum Master 26](#_Toc103260769)

[2.6.4. Scrum Team Members 27](#_Toc103260770)

[2.7. Project Initiation 27](#_Toc103260771)

[2.7.1. Project Charter 27](#_Toc103260772)

[2.7.1.1. Project Vision 27](#_Toc103260773)

[2.7.1.2. Project Mission 28](#_Toc103260774)

[2.7.1.3. Project Success Criteria 28](#_Toc103260775)

[2.7.2. Use Case Diagram & Report 28](#_Toc103260776)

[2.7.3. Prioritized Use Cases 28](#_Toc103260777)

[2.7.4. Epics 29](#_Toc103260778)

[2.8. Product Backlog – Features 29](#_Toc103260779)

[2.8.1. User Story Map 30](#_Toc103260780)

[2.8.2. Prioritized User Stories 30](#_Toc103260781)

[2.9. Project Deliverables – Release Planning 30](#_Toc103260782)

[2.10. Burnout Chart/Line 31](#_Toc103260783)

[Bibliography 32](#_Toc103260784)

[Figure 1 BPMN Business Model Plan Covid Tracker Process Oriented 2](#_Toc101204209)

[Figure 2 Product Backlog Topics Summary 4](#_Toc101204210)

[Figure 3 Project's Persona 7](#_Toc101204211)

[Figure 4 General Dev Mgmt. Overview 10](#_Toc101204212)

[Figure 5 Main Mobile System Use Case 11](#_Toc101204213)

[Figure 6 UML Class Diagram με cardinalities/πληθυκότητες 12](#_Toc101204214)

[Figure 7 Log in Screen – Wireframe 14](#_Toc101204215)

[Figure 8 Dashboard – Wireframe 15](#_Toc101204216)

[Figure 9 Add Patient UI – Wireframe 16](#_Toc101204217)

[Figure 10 Statistics Viz - Wireframe 17](#_Toc101204218)

[Figure 11 Basic Flowchart Android App 18](#_Toc101204219)

[Figure 12 State Machine Basic 18](#_Toc101204220)

[Figure 13 Site Map 19](#_Toc101204221)

[Figure 14 Main Dashboard Heat Areas (red is more eye tracked) 20](#_Toc101204222)

[Figure 15 Process Management Lifecycle 21](#_Toc101204223)

[Figure 16 ITSM Identification 22](#_Toc101204224)

[Figure 17 ITSM Initiation 22](#_Toc101204225)

[Figure 18 PERT timeline \*no critical path (CPM) for one-way-trip path 23](#_Toc101204226)

[Figure 19 Work Breakdown process overview 23](#_Toc101204227)

[Figure 20 Scrum Canvas 24](#_Toc101204228)

[Figure 21 SAFe - Enterprise Framework 24](#_Toc101204229)

[Figure 22 Scrum Cabinet 25](#_Toc101204230)

[Figure 23 User Story Map 31](#_Toc101204231)

[Figure 24 Burndown chart Sprint 1 Y as Story points of Sprint No.1 == 30 task points 32](#_Toc101204232)

[Table 1 Stakeholder List 25](#_Toc101002881)

[Table 2 Product Owner 25](#_Toc101002882)

[Table 3 Scrum Master 26](#_Toc101002883)

[Table 4 Scrum Team Members 27](#_Toc101002884)

[Table 5 Prioritized Use Cases 28](#_Toc101002885)

[Table 6 Use Case to Epic Step 29](#_Toc101002886)

[Table 7 Prioritized User Stories 30](#_Toc101002887)

[Table 8 Project Deliverables 31](#_Toc101002888)

[Table 9 Release Configuration 31](#_Toc101002889)

# agenda

## Brief

* Project Name:
  + Augmented Reality in Mobile with Unity
* Project Description:
  + Ένας πελάτης επιθυμεί να αναπτύξει και να δοκιμάσει μια εφαρμογή επαυξημένης πραγματικότητας για ψυχαγωγία κάνοντας score στο Dartboard.
* Για ποιο είναι?
  + End-User – αυτό το προϊόν είναι για την ψυχαγωγία χρηστών κάνοντας score σε Darboard.
* Feature List (product requirements)
  1. GPS (*long + latt + altitude*)
  2. Database Store & Retrieve (*CRUD*)
  3. Web Service Access: Unity Ad Network
  4. Augmented Reality SDK
  5. Access to phone Resources (*manifest.xml*)
     1. Camera
     2. Location Awareness (*Fine position*)
* Deliverables
  + Wireframes για client approval
  + High fidelity prototype του product
  + User Testing
  + Usability Report
  + UI Assets για developers
* Timeline/Deadline
* Η διάρκεια περιγράφεται παρακάτω ως subtask

## Description Scope

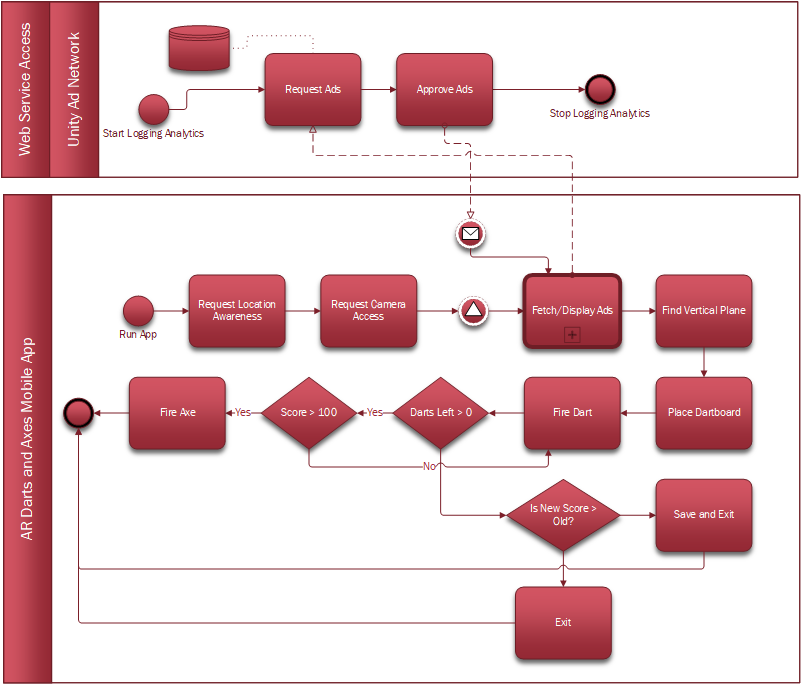


Figure 1 BPMN Business Model Plan AR Darts and Axes Process Oriented

Ο σκοπός αυτού του εγγράφου είναι να παρέχει έγκυρες πληροφορίες για τα μέλη του, συμπεριλαμβανομένης της σχεδίασης υψηλού επιπέδου & χαμηλού (*HLD & LLD*) σε μια συνοπτική εφαρμογή της εφαρμογής Περιεχόμενο Επαυξημένης πραγματικότητας για ψυχαγωγίας παιχνιδιού Darts and Axes. (Wikipedia, 2021)

Η εφαρμογή θα αποτελείται από λειτουργικότητα διεπαφής frontend & backend και διεπαφές εφαρμογών για κινητά. Θα επιτρέψει στον χρήστη να βρίσκει επίπεδα στο χώρο τοποθετώντας το Dartboard ώστε να αρχίσει το παιχνίδι καθώς θα υπάρχει επικοινωνία με geolocation, & web service access και αποθηκεύσει νέου σκορ & perks ξεκλειδώματα για εναλλαγή χαρακτηριστικών. (www.bpmn.org, n.d.)

## Financial Scope

Το έργο θα προσφέρει έσοδα μέσο του Ad network Unity χωρίς mediation.

* Analytics
* Διαχείριση κερδών

## Focused on Delivery

*Subtasks/activities:*

1) Requirement’s analysis

2) Database/Business Entities design

3) Interface design

4) Database/ Business Entities development και data entry

5) Interface development

6) System completion

7) Application installation και control (integration)

8) Support Q&A

## Project Execution

Μια βάση εξειδίκευσης του προσωπικού και διαθέσιμη υποδομή είναι η επιλεγμένη προσέγγιση ευέλικτης ομάδας scrum. Η εγγραφή και διαχείριση του κύριου μαθήματος θα γίνει μέσο συνεργασίας και ευέλικτης πλατφόρμας JIRA με σύστημα παρακολούθησης και καταγραφής προβλημάτων/πηγή εισιτηρίων και έλεγχος έκδοσης DVCS GIT.

Product Backlog Overview

Παρακάτω ακολουθεί μια αρχική εκτίμηση η οποία θα αλλάξει δυναμικά (ενημέρωση / πέρασμα του χρόνου) με περισσότερες πληροφορίες καθώς θα αλλάξουν οι προτεραιότητες καθώς και οι απαιτήσεις του έργου ενδέχεται να αλλάξουν.



Figure 2 Product Backlog Topics Summary

\*Από την παραπάνω ενότητα Ημερήσια διάταξη συνοψίζονται τα πάντα, συμπεριλαμβανομένου του Gantt και backlog, αυτά δεν είναι καθόλου κοντά στις πραγματικές απαιτήσεις αλλά μια στενή γενικευμένη εκτίμηση του τι πρέπει να αναμένεται. Στη συνέχεια θα εισαχθούν πραγματικές υλοποιήσεις σε αυτό το έγγραφο.

## Technologic Approach

Η εφαρμογή θα είναι mobile based

1. Augmented Reality in Unity Game Engine
   1. C# Scripting
2. Location Awareness
3. Database SQLite (CRUD)
4. Web Service Access (Unity Ads Network)
5. Android Phone resources access (manifest.xml)

# DESIGN

## PROBLEM ANALYSIS

Πρόβλημα:

«Να γίνει ο σχεδιασμός για μια εφαρμογή Android για κινητό, η οποία θα χρησιμοποιεί τουλάχιστον συνδυασμό 5 τεχνολογιών:

* Web-Service access
* Location Awareness
* Store and retrieve data (Create, Read, Update, Delete)
* Accelerometer
* Embedded applications
* Augmented Reality
* QR Coding Techniques
* Sensor Technology
* Access to phone resources (i.e., camera, address book etc.)

Την εφαρμογή θα την χειρίζεται ένας χρήστης για ψυχαγωγία. Σε αυτή θα μπορεί να παίξει Darts σε επαυξημένη πραγματικότητα καθώς και το νέο σκορ θα καταγράφεται άμα υπερβαίνει το παλιό καθώς θα υπάρχει και σύστημα διαφημίσεων για web service access.»

Σύνοψη Λύσης:

Μια εφαρμογή επαυξημένης πραγματικότητας για κινητό συγκεκριμένα Android Platform θα χρειαστεί μία Game Engine καθώς και πλούσιο οικοσύστημα με ένα Framework για AR back-end service & Controllers και μία βάση δεδομένων για Non-volatile αποθήκευση δεδομένων που σημαίνει ότι η εφαρμογή θα λειτουργεί σαν front-end θα την χειρίζεται ο εκάστοτε χρήστης βρίσκοντας ένα κάθετο πλάνο κατόπιν τοποθετώντας το Dartboard ενώ το νέο σκορ θα εισάγεται μετά από αίτημα στο back-end service ώστε να εκχωρηθεί άδειας αποθήκευσης τους σε μόνιμα μέρη αποθηκεύσεις δομής.

Η φυσική υποδομή θα πρέπει να λάβει χώρα τοπικά στο κινητό καθώς να τηρηθεί ή μη-αμελητέα αποθήκευση δεδομένων για να υπάρξει Confidentiality, availability, reliability, Integrity λόγο privacy issues όπως access to phone resources.

Η χρήση τεχνολογιών αναφέρεται εδώ -> Technologic Approach

## Μεθοδολογία

Επιλογή Οικοσυστήματος & SDK:

### Η Επιλογή ενός SDK πρέπει να γίνει με βάση τι οικοσύστημα που το περιβάλλει. Μέσα από πληθώρα επιλογών όπως Android Studio, Unreal Engine, Unity, η καλύτερη επιλογή αναδείχτηκε η Unity γιατί παρέχει ένα δικό της non-license framework High Level Interface Subsystem (ARFoundation) το οποίο είναι abstracted (interface & not implementation) για την δημιουργία είτε σε Android είτε σε IOS AR app – Code Once Run Everywhere. Οι υπόλοιπες επιλογές θα ήταν κατευθείαν προγραμματισμός σε Low level SDK όπως ARCore για Android και ξεχωριστό κώδικα για ARKit IOS αλλά λόγο της νέας τεχνολογίας και συνεχείς ενημερώσεις δεν είναι πολύ ώριμα τα framework από άποψη αρχιτεκτονική και επεκτασιμότητας αλλά είναι σε ερευνητικό πειραματικό στάδιο πράγμα που δημιουργεί «σπάσιμο του κώδικα» ακόμα και μέσα από ένα Minor update του εκάστοτε SDK. Για αυτό το λόγο το Framework ARFoundation της Unity που είναι abstracted είναι ένα Interface που ο τελικός developer αρκεί να φτιάξει μόνο ένα κώδικα και η ίδια η unity μετά αναλαμβάνει τα Low level χαρακτηριστικά πως και τι μετά από κάθε update.

Compile:

Η Εφαρμογή περιέχει False Positives ως προς warnings/errors compilation λόγο Bugs της Engine ωστόσο γίνεται compile/build κανονικά διότι όπως αναφέραμε είναι False Positives σύμφωνα με την Unity Bug Report System. Επίσης αξίζει να τονιστεί ξανά ότι η εφαρμογή έχει γίνει publish στο Google Play Store εφόσον πέρασε όλους τους standard Quality Service ελέγχους (Q&A).

Αρχιτεκτονική κώδικα:

Χρησιμοποιούνται Design Patterns Singleton & Observer δηλαδή για 1 spawn Instance μιας κλάσεως σε ένα game object μόνο και events/Dispatcher Listeners on Bind για trigger των subscribers από τον Broadcast.

Παραδείγματα Singleton:

* Dartboard
* Boss fight
* SoundManager
* AppManager
* Database Connector
* Ads Network Initializer
* GPSController
* Scoring System



Figure Unity C# Script Snippet Singleton Pattern

Παράδειγμα Observer:

Observer χρησιμοποιούνται για Notify ενός event έτσι αποφεύγουμε το διαρκεί έλεγχο μέσο Update Per Frame Checking (on Update() || Tick()).



Figure Unity C# Script Snippet Observer Pattern

## PERSONA

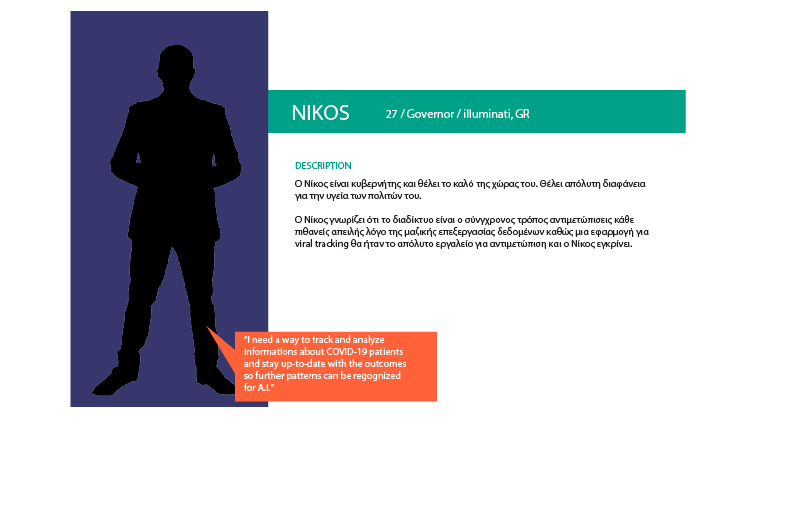


Figure 5 Project's Persona

Το project θα επεκταθεί όπως θέλει ο Νίκος (Faller, 2019).

## Requirement Analysis (non/functional) & Use Cases Ερμηνείες

indicative specification functional requirements

**Function Name – Use Case:** Account Login

**Short Description:** Ο Υπάλληλος του υπουργείου προστασίας του πολίτη θα συνδέεται στο σύστημα.

**Conditions:** Μόνο έγκυρη λογαριασμοί από το κατάλληλο domain/department.

**Inputs:** Front-end portal authenticated στο back-end

**System behavior – Main Flow:** Το σύστημα λαμβάνει Δεδομένα από το mobile app συγκρίνοντας κάθε μοναδικό χρήστη ξεχωριστά με βάση τα πεδία εισαγωγής του. Ο «χρήστης» ζητά ένα διακριτικό στην ασφάλεια Spring, εάν παρείχε σωστό ένα JWT που επιστρέφει για να αποκτήσει πρόσβαση στη λειτουργικότητα RESTful API to-do.

**Alternative Flow:** -

**Output – Exit Status:** Πρόσβαση σε χρήση Services του συστήματος.

**Function Name – Use Case:** Accounts & ΑΑΑ (*authentication, authorization, accounting*)

**Short Description:** Πολλαπλοί είδη λογαριασμού Role-based access control (*RBAC*).

**Conditions:** Έλεγχος και έναρξη καταγραφής επιπέδου πρόσβασης και δραστηριότητας κατά την σύνδεση.

**Inputs:** Front-end portal authenticated στο back-end.

**System behavior – Main Flow:** Κατά την είσοδο στο σύστημα ελέγχεται το επίπεδο πρόσβασης στο back-end με βάση τα εισαγόμενα στοιχεία χρήστη που αναθέτη ένα RBAC αριθμό καθώς και services για AAA enabled.

**Alternative Flow:** -

**Output – Exit Status:** Διαφοροποίηση & Έλεγχος πρόσβασης σε χρήση Services του συστήματος.

**Function Name – Use Case:** Patient Registry

**Short Description:** Ο συνδεμένος χρήστης διαχειρίζεται νέους/παλιούς ασθενής.

**Conditions:** Καταχώρηση ασθενών εφόσον βγουν θετικοί μέσο τεστ.

**Inputs:** Mobile user interface

**System behavior – Main Flow:** Ο χρήστης εισάγει με βάση το RBAC του πληροφορίες στην βάση δεδομένων με βάση τους ασθενής και περίοδο νοσήσεις.

**Alternative Flow:** -

**Output – Exit Status:** Καταχώρηση εγγραφών χρήστη και ενημέρωση των στοιχείων/εγγραφών τους.

**Function Name – Use Case:** Export Statistical analysis.

**Short Description:** Ο διαχειριστής εισάγει ή αφαιρεί μια καταχώρηση από το σύστημα.

**Conditions:** Database <> {} and patient <> [].

**Inputs:** Mobile user interface.

**System behavior – Main Flow:** Το σύστημα παράγει αντλώντας δεδομένα τις βάσεις δεδομένων σε δομή δεδομένων και τα σκιαγραφεί εικονικά μετέπειτα από αλγοριθμική ανάλυση τους στην εφαρμογή κινητού για την πορεία, πλήθος, κ.α., περί νοσήσεις.

**Alternative Flow:** -

**Output – Exit Status:** Statistical visualization output για Data analysis.

**Function Name – Use Case:** Admin CRUD’s Database.

**Short Description:** Ο διαχειριστής εισάγει ή αφαιρεί μια καταχώρηση από το σύστημα.

**Conditions:** RBAC login Admin Level.

**Inputs:** Front-end page admin account.

**System behavior – Main Flow:** Το σύστημα είναι σε θέση να αφαιρέσει ή να προσθέσει μια νέα καταχώρηση με βάση τις ενέργειες λειτουργίας διαχειριστή.

**Alternative Flow:** -

**Output – Exit Status:** Business entity database update.

indicative specification non-functional requirements

**Function Name:** CIA (Confidentiality, integrity and availability) triad

**Function Name:** Optimize Performance with Profiler

**Function Name:** Maximize Battery Life

**Function Name:** Optimize for Responsiveness

**Function Name:** Improve App Security (SQLite αντί public database λόγο ενός σκορ και μόνο χρήστη)

**Function Name:** Testing

Για να επιτευχθούν τα NFR (non-functional requirements) χρειάζεται να γίνουν τα «αυτονόητα» από μεριά χρήστη δηλαδή μια ομαλή θετική αδιάκοπη εμπειρία.

Από τεχνικής άποψης να υπάρχουν γενικά οι παρακάτω λειτουργείες σε ομαλή φόρμα:

* Performance
* Availability (παίζοντας με no internet required λόγο ads)
* Scalability
* Usability
* Security
* Modifiability
* Maintainability and testability
* Cost

Όλα αυτά πρέπει να σχεδιαστούν proactively στο κομμάτι της σχεδίασης ώστε στην αρχιτεκτονική του προγράμματος να υπάρχουν σωστές βάσεις και Design Patterns για βελτιστοποίηση κώδικα αλλά να ενώνονται και με τεχνολογίες που επιτρέπουν να λύσουν τα παραπάνω προβλήματα όπως η χρήση Industry Standard SDK, tools για την δόμηση του προγράμματος είναι το ποιο strategic smart decision από την αρχή δηλαδή χρήση JDK/SDK (*latest version*) (*άμα φυσικά αυτό είναι εφικτό σύμφωνα και με τα υπόλοιπα υποσυστήματα που θα πρέπει να επικοινωνεί το σύστημα*) Version Control System (*i.e., git -> GitHub*), profiling/monitoring tools, efficient data structures.

## UML Representation Analysis

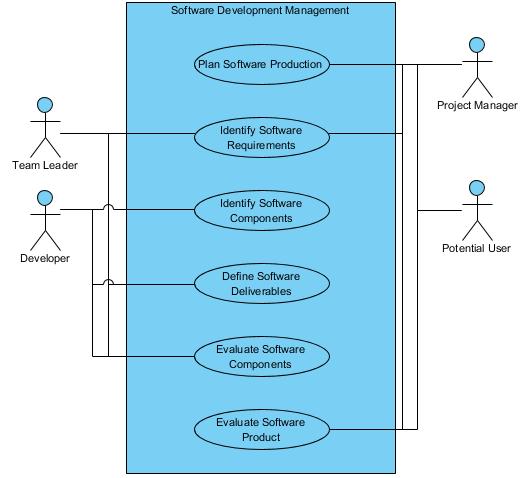


Figure 6 General Dev Mgmt. Overview

## Systems Use Case

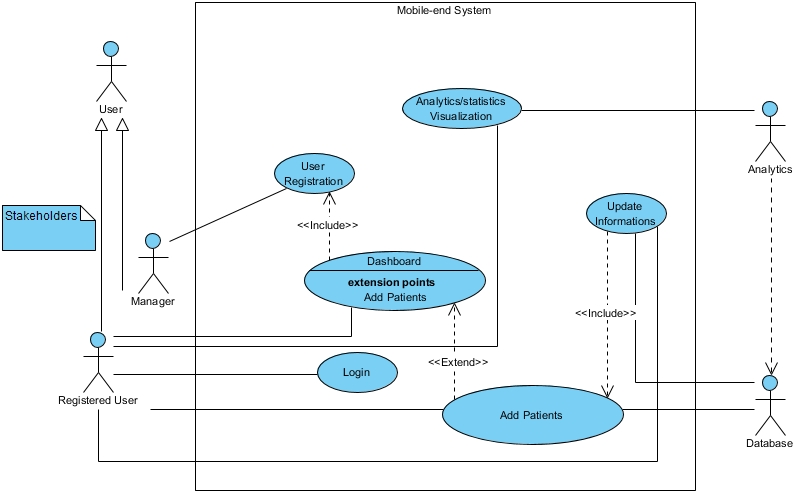


Figure 7 Main Mobile System Use Case

## Class Diagrams & ERD

Λόγο χρήσης ORM (*object relational mapping*) τεχνολογιών το παρακάτω διάγραμμα μπορεί να χρησιμοποιηθεί και ως ERD. (hibernate.org, n.d.; Visual-paradigm.com, 2011; Paradigm, 2019)

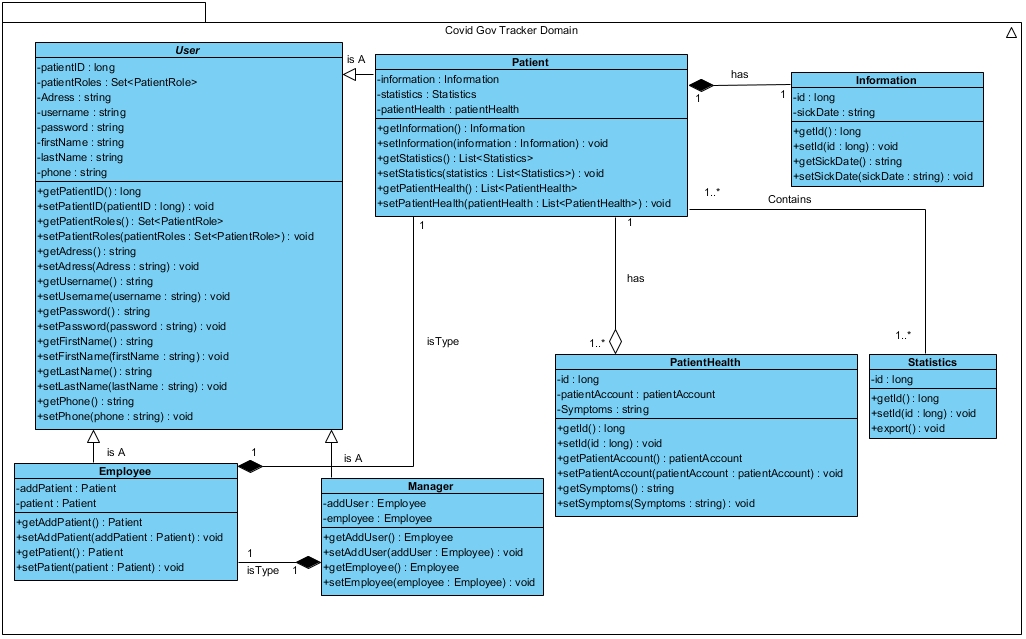


Figure 8 UML Class Diagram με cardinalities/πληθυκότητες

Στην παραπάνω λίστα κλάσεων όλοι είναι και patients έστω και με empty history, η class User είναι abstract που σημαίνει δεν δημιουργούμε objects από αυτήν μόνο κληρονομούμε (Oracle, 2019).

Ο Admin δεν υπάρχει σαν κλάση διότι είναι RBAC 0 (*Role Based Access Control*) και κάνει Login μέσω web browser να προσθέσει Managers. (docs.oracle.com, n.d.)

Οι Manager είναι ταυτόχρονα και employee όταν δημιουργείται δηλαδή ένας Manager ⬄ Employee έτσι έχει πρόσβαση σε όλες τις μεθόδους ενός employee, επίσης αφού είναι και patient αυτό του δίνει πρόσβαση στην statistics που είναι μια List/Sequence από όλους τους ασθενής (*εφόσον την εφαρμογή την χειρίζεται μόνο εξειδικευμένο προσωπικό και δεν είναι για όλους άρα μπορούμε με ασφάλεια να δηλώσουμε μεθόδους και στην κλάση Patient που θα ήταν πιο high Privileged*).

Οι employees προσθέτουν και κάνουν Update άλλους Patients εφόσον έχουν πλήρη πρόσβαση λόγο composition στην class patient θέτοντας CRUD requests προς τον back-end service server controller.

Ένας patient έχει πολλά patient health διότι είναι incremental/record κάθε περίπτωση νοσήσεις διότι μπορεί να νοσήσει πολλαπλές φορές άρα θέλουμε δομή record και όχι όλα σε 1 προς 1 αλλά 1 προς πολλά από άποψη patient => patientHealth, επίσης ο patient μπορεί να υπάρχει αλλά δεν σημαίνει ότι έχει νοσήσει λόγο την ιεραρχία του συστήματος ότι ένας manager & User είναι και patient έστω και με empty history άρα χρειαζόμαστε να μην είναι δεσμευμένο το History ⬄ patientHealth που σηματοδότη την νοσήσει.

Η κλάση statistics έχει μεθόδους και properties-objects τύπου List<Patient> get & set (*λόγο many-to-many*) ωστόσο στο σχεδιάγραμμα δεν φαίνεται διότι θελήσαμε να δηλώσουμε την αποφύγει json loop έτσι παραλήφθηκε η δήλωση τους. Επίσης λόγο “double” composition μπήκε το πιο general class relationship “association”.

## Storyboards & Wireframes (low fidelity)

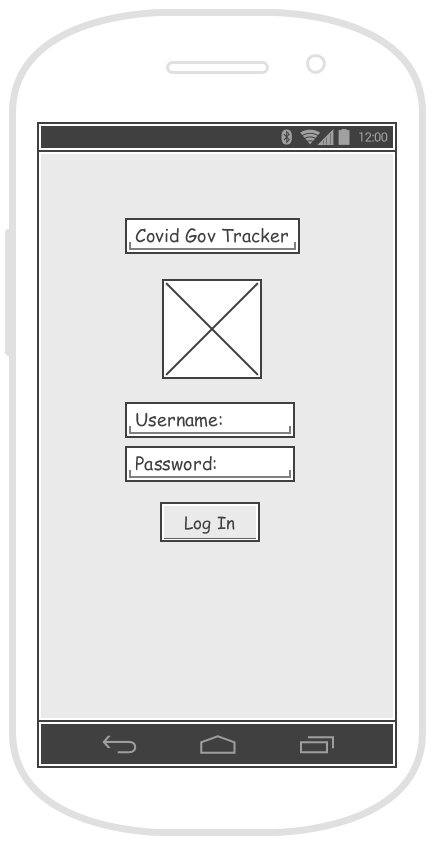


Figure 9 Log in Screen – Wireframe

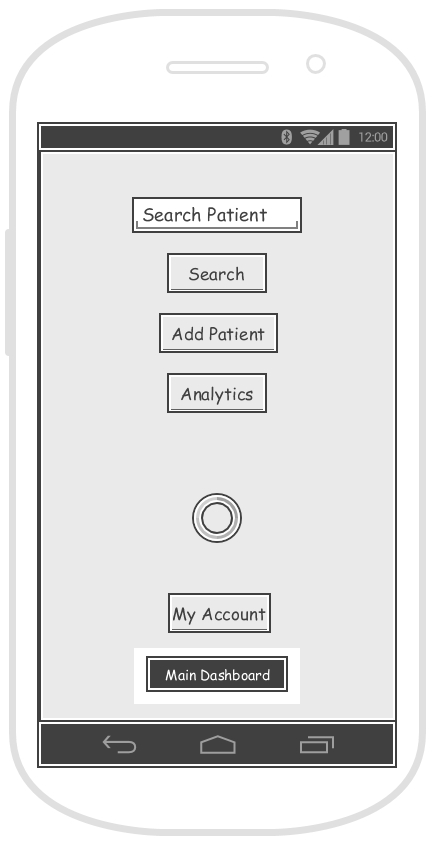


Figure 10 Dashboard – Wireframe



Figure 11 Add Patient UI – Wireframe

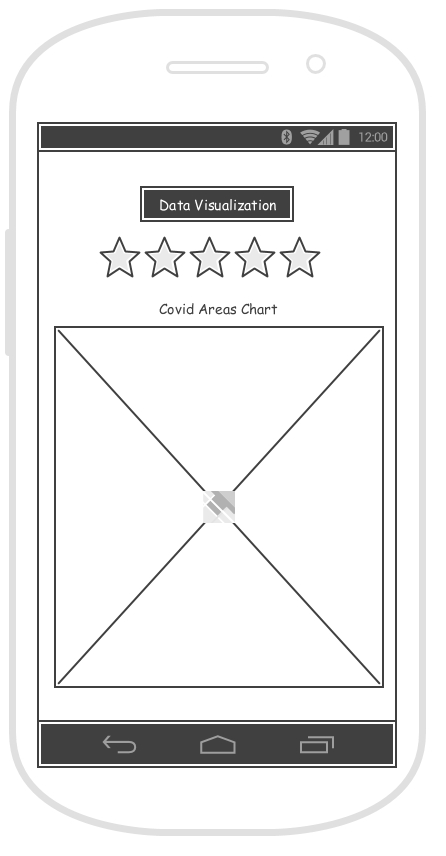


Figure 12 Statistics Viz - Wireframe

## User Flows

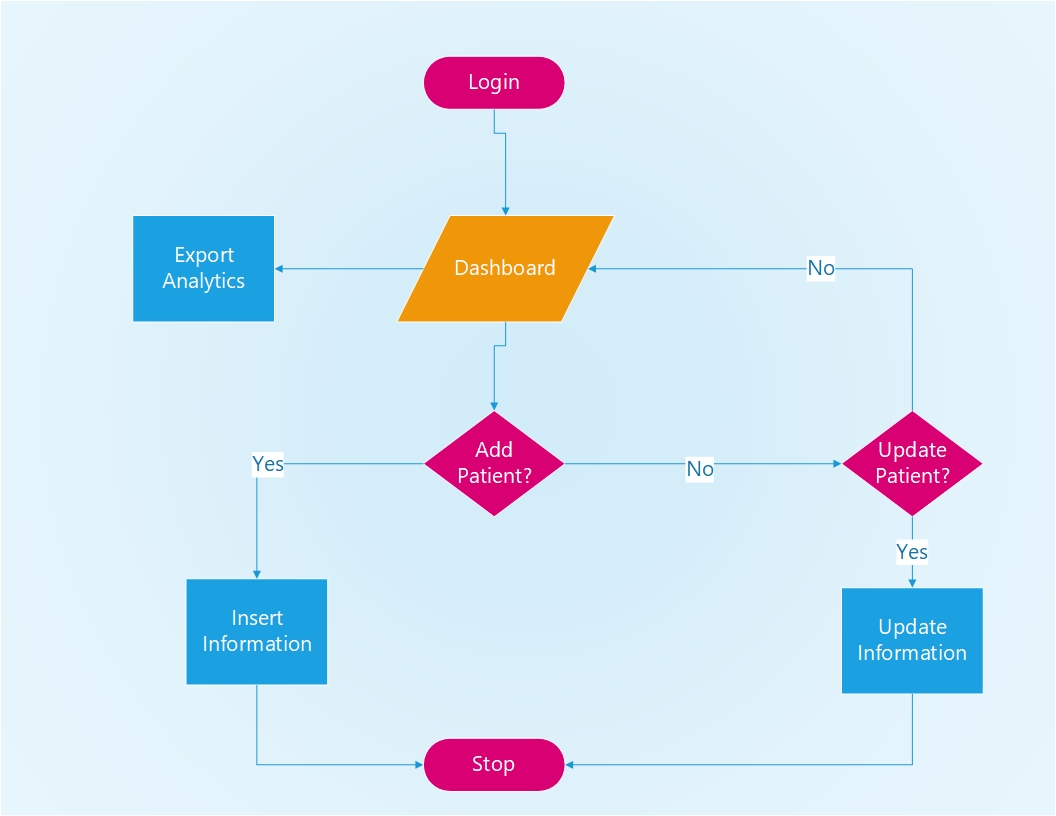


Figure 13 Basic Flowchart Android App

## State Machine/State Chart Diagram

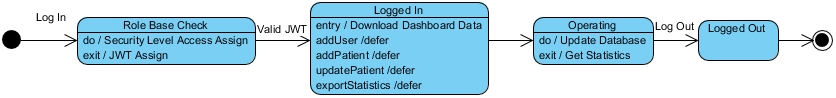


Figure 14 State Machine Basic

## Testing

Για την σωστή διεκπεραίωση των λειτουργειών εφαρμογής θα χρησιμοποιηθεί η τακτική Test Driven Development (*TDD*) στην οποία πρώτα γράφεις των κώδικα του testing και τι κάνεις σαν expect και ύστερα γράφεις των κώδικα του implementation σε συνδυασμό με ένα CI/CD e.g., Jenkins θα μπορεί να πραγματοποιηθεί σε κάθε commit GitHub real-time checking άμα έγινε αποδεκτή η εφαρμογή στα testing πριν/μετά για σωστό deploy σε containers Kubernetes, docker.

Στην συγκεκριμένη τεχνολογία Spring Boot το testing γίνεται μόνο μέσω χρήσης Dependency Injection ώστε να επιτραπεί το Unit testing καθώς και loggers τόσο στο front-end (native android) όσο και στο back-end (spring).

# Project Management

## Lifecycle

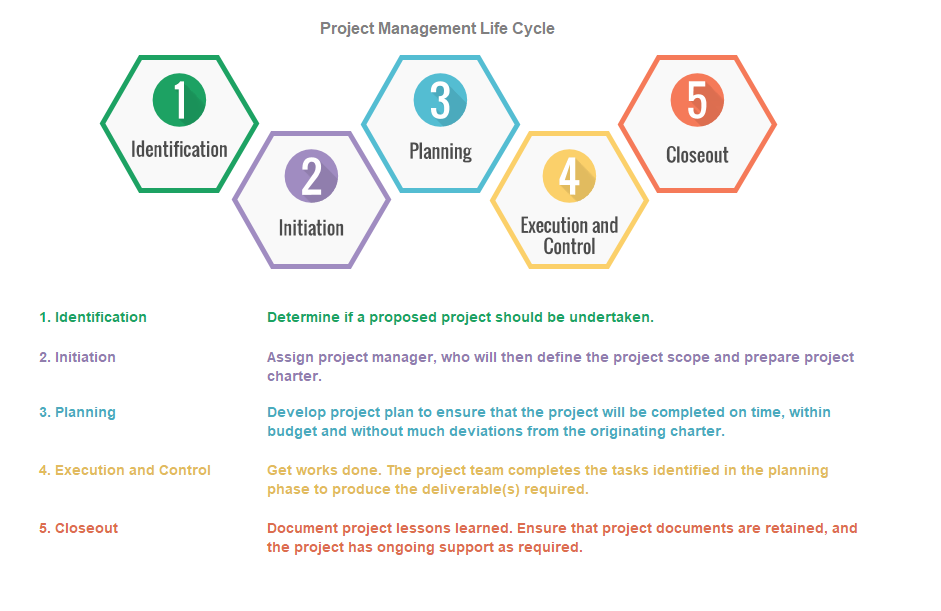


Figure 15 Process Management Lifecycle

## Task/Time Schedule PERT

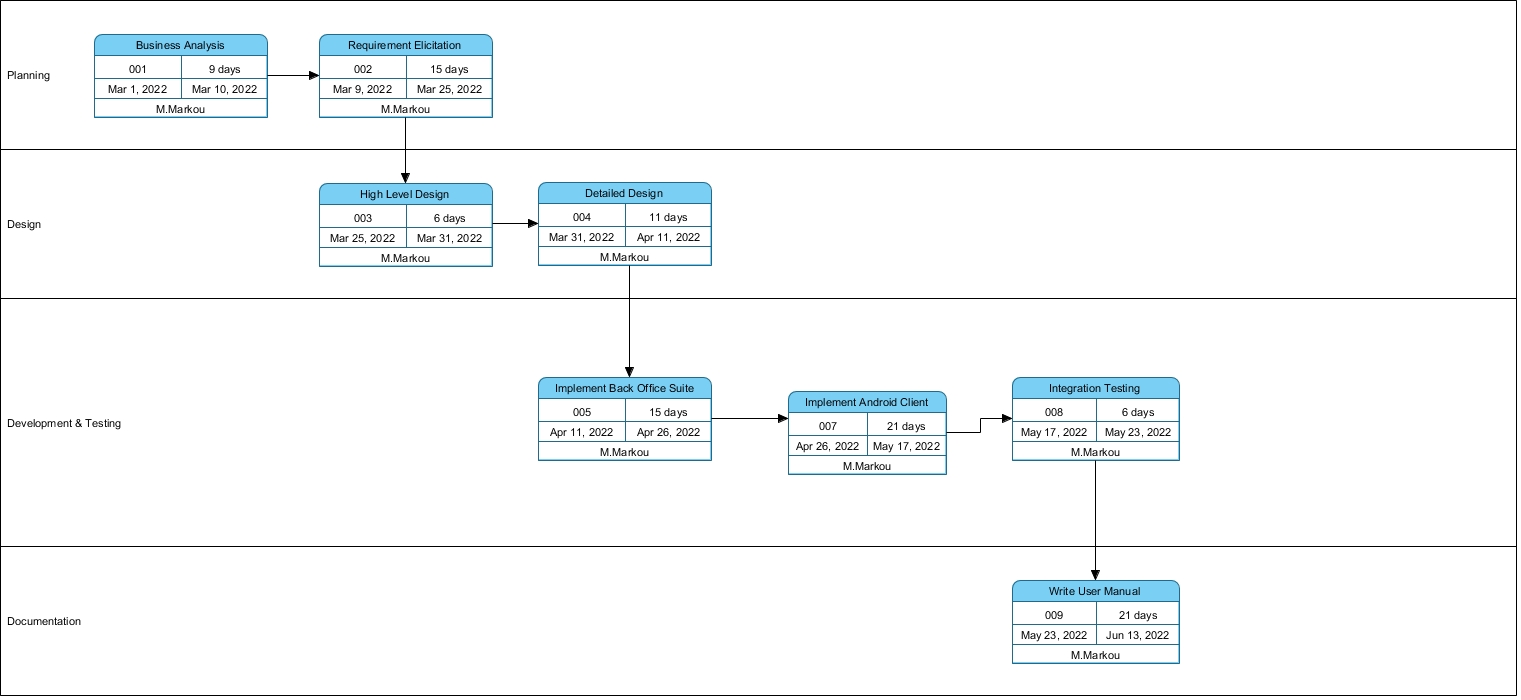


Figure 16 PERT timeline \*no critical path (CPM) for one-way-trip path

## Work Breakdown Structure (WBS)

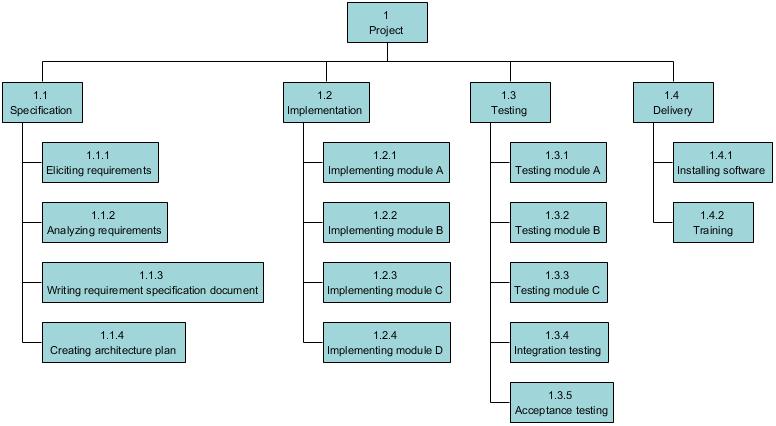


Figure 17 Work Breakdown process overview

## Sprint Canvas

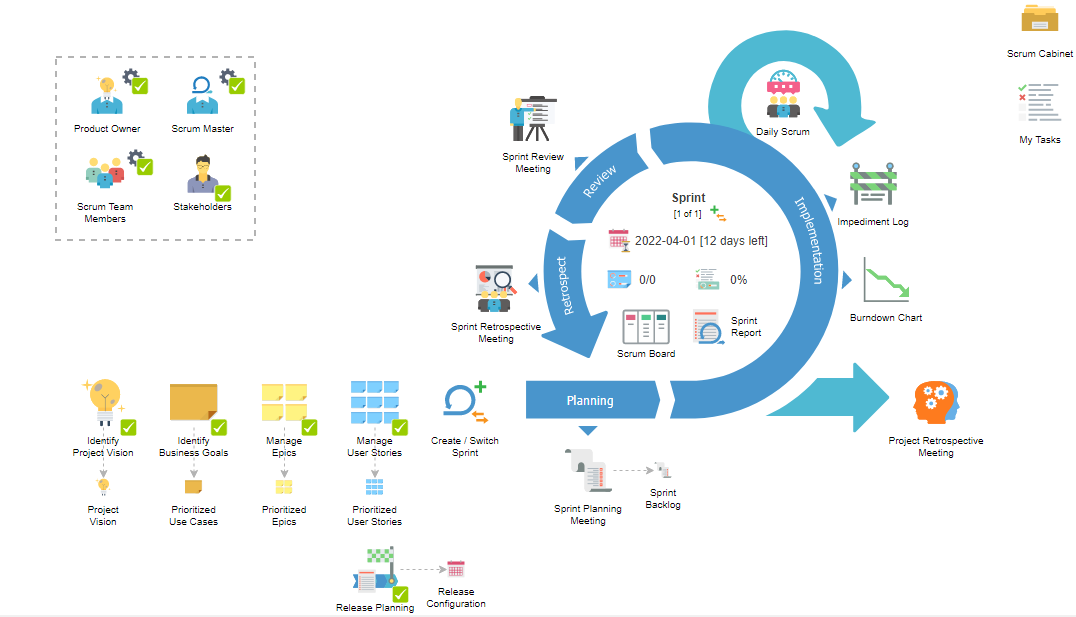


Figure 18 Scrum Canvas

(Atlassian, 2019; Scrum.org, 2019)



Figure 19 SAFe - Enterprise Framework

## Scrum Cabinet Reports

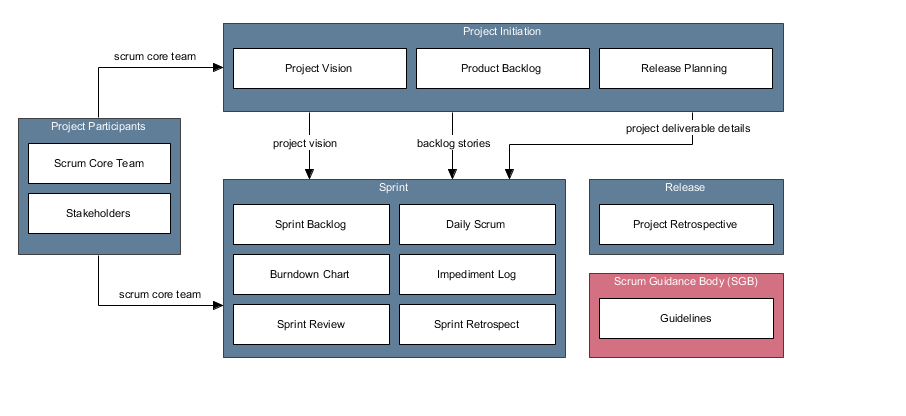


Figure 20 Scrum Cabinet

## Project Participants – κατανομή εργασιών

* + 1. Stakeholder List

| Name | Role | Responsibilities |
| --- | --- | --- |
| Michail Markou | Sponsor, Customer, User | Διαχειρίζεται το τελικό αποτέλεσμα και το όραμα του έργου |

Table 1 Stakeholder List

* + 1. Product Owner

| Member | Michail Markou |
| --- | --- |
| Responsibilities | * Καθορίζει το όραμα του έργου * Βοηθά στη δημιουργία του Χάρτη του Έργου * Προσδιορίζει τα ενδιαφερόμενα μέρη του έργου * Βοηθά στον προσδιορισμό των μελών της ομάδας scrum * Δημιουργήστε Epics * Δημιουργήστε, ορίστε και δώστε προτεραιότητα σε ιστορίες χρηστών * Δημιουργήστε σχέδιο έκδοσης και κρατήστε το ενημερωμένο * Έγκριση ιστοριών χρηστών * Εξηγεί τις ιστορίες των χρηστών και διευκρινίζει τις απαιτήσεις για την ομάδα scrum στη συνάντηση προγραμματισμού sprint * Παρέχει καθοδήγηση και διευκρίνιση στην εκτίμηση της προσπάθειας για εργασίες * Grooms prioritized backlog * Αποδοχή/απόρριψη παραδοτέων/deliverables * Παρέχετε σχόλια στις ομάδες scrum master και scrum team * Βοηθήστε την ανάπτυξη εκδόσεων προϊόντων και συντονίστε το με τον πελάτη * Συμμετοχή σε έντυπη αναδρομική συνάντηση |

Table 2 Product Owner

* + 1. Scrum Master

| Member | Michail Markou |
| --- | --- |
| Responsibilities | * Βοηθά στον εντοπισμό των ενδιαφερομένων στο έργο * Διευκολύνει τη συγκρότηση της ομάδας scrum * Διευκολύνει τη δημιουργία επών * Βοηθά τον ιδιοκτήτη του προϊόντος να δημιουργήσει και να διατηρήσει το product backlog * Συντονίζει τη δημιουργία του Σχεδίου Έκδοσης * Βοηθά στη δημιουργία και τον καθορισμό ιστοριών χρηστών * Διευκολύνει τις συναντήσεις * Διευκολύνει την ομάδα scrum στη δημιουργία εργασιών για το επόμενο sprint * Διευκολύνει την ομάδα scrum στην εκτίμηση προσπάθειας * Υποστηρίζει την ομάδα scrum στη δημιουργία παραδοτέων * Βοηθά στη διατήρηση του αρχείου καταγραφής εμποδίων/impediment log * Διασφαλίζει ότι ανακαλύπτονται και επιλύονται ζητήματα που επηρεάζουν την ανάπτυξη |

Table 3 Scrum Master

* + 1. Scrum Team Members

| Member | Responsibilities |
| --- | --- |
| Michail Markou | * Supervisor & Senior Programmer * Documentation Writer & Reviewer |
| John Chaidos | * Junior Programmer |
| Manios Athanasios | * Documentation Writer & Reviewer |
| Alantzopoulos | * Documentation Writer |

Table 4 Scrum Team Members

## Project Initiation

* + 1. Project Charter
       1. Project Vision

Σύστημα για χρήση ανίχνευσης καταγραφής κρουσμάτων καθώς και στατιστικής ανάλυσης των δεδομένων για έρευνες αλλά και πορεία κρατικής υγείας. Με ευκολία χρήσεις από mobile devices.

* + - 1. Project Mission

Διευκόλυνση καταγραφής κρουσμάτων και σύγκριση παλιών με νέων καθώς και στατιστική πορεία της χώρας. Ταυτόχρονη διευκόλυνση με on-demand Patient Registry μέσο κινητού. Stay up-to-date.

* + - 1. Project Success Criteria
* Έγκυρη καταγραφή κρουσμάτων και real-time update κατάστασης
* Στατιστική ανάλυση πορείας δεδομένων και πρόληψης ιογενών περιοχών
* Μείωση των κρουσμάτων με καθοδήγηση ασθενών και μη.
  + 1. Use Case Diagram & Report

(βλ. Figure 5)

* + 1. Prioritized Use Cases

| Name | Description | Priority | Size | Complexity |
| --- | --- | --- | --- | --- |
| Dashboard | Κεντρικό μενού ελέγχου ενεργειών | Must | Medium | Medium |
| User Registration | Προσθήκη χρήστη στο σύστημα | Must | Large | High |
| Login | Σύνδεση χρήστη στο σύστημα | Must | Large | Medium |
| Analytics/statistics Visualization | Γραφική αναπαράσταση πληροφοριών και απόκτηση γνώσεις | Should | Very Large | High |
| Update Information’s | Ενημέρωση κατάστασης ασθενών | Must | Medium | Low |

Table 5 Prioritized Use Cases

* + 1. Epics

| Name | Parent Use Case | Priority | Risk |
| --- | --- | --- | --- |
| Registered User add Patients in the Database | Add Patients | Must | Low |
| Export analytics to User Interface | Analytics/statistics Visualization | Should | High |
| User RBAC System Access to resources | Login | Must | Low |
| User Updates Information about Patients | Update Information’s | Must | Medium |
| User Main Navigation to system resources | Dashboard | Must | Medium |

Table 6 Use Case to Epic Step

## Product Backlog – Features

* + 1. User Story Map

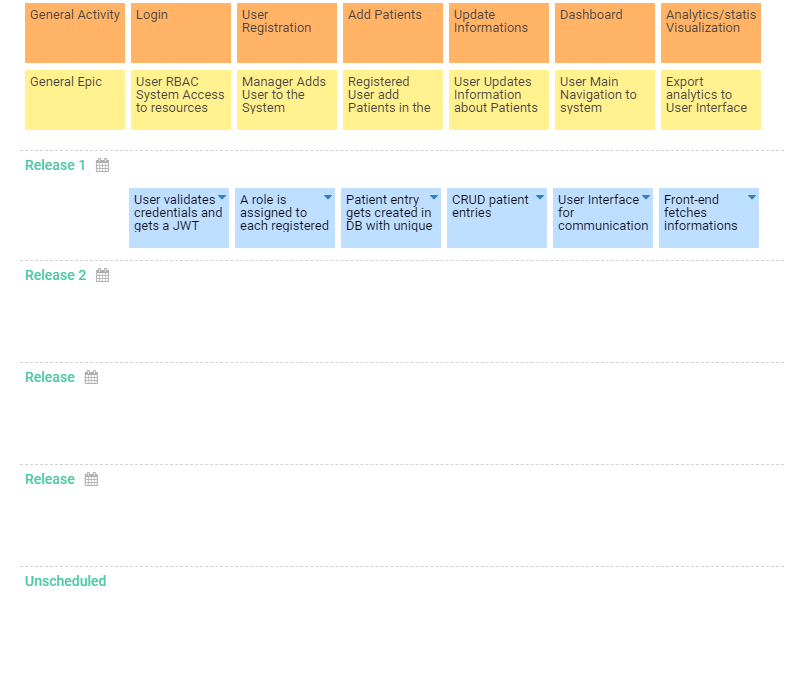


Figure 21 User Story Map

* + 1. Prioritized User Stories

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Epic** | **Status** | **Story Points** | **Priority** |
| A role is assigned to each registered user | Manager Adds User to the System | Approved | 3 | Must |
| User validates credentials and gets a JWT from backend | User RBAC System Access to resources | Approved | 5 | Must |
| Patient entry gets created in DB with unique ID | Registered User add Patients in the Database | Approved | 3 | Must |
| CRUD patient entries | User Updates Information about Patients | Approved | 1 | Must |
| User Interface for communication to back-end | User Main Navigation to system resources | Approved | 3 | Must |
| Front-end fetches information’s through back-end from the database to display knowledge | Export analytics to User Interface | New | 8 | Should |

Table 7 Prioritized User Stories

## Project Deliverables – Release Planning

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Deliverable | Description | Priority | Status | Owner |
| Covid Gov Tracker (Android) | The Android Version of Front-end communication to the system | High | In Progress | Michail Markou |

Table 8 Project Deliverables

|  |  |  |  |
| --- | --- | --- | --- |
| Release | Description | Release | Description |
| Release 1 | Support Major Features for Patient Registry and Tracking | Release 1 | Support Major Features for Patient Registry and Tracking |

Table 9 Release Configuration

## Burnout Chart/Line

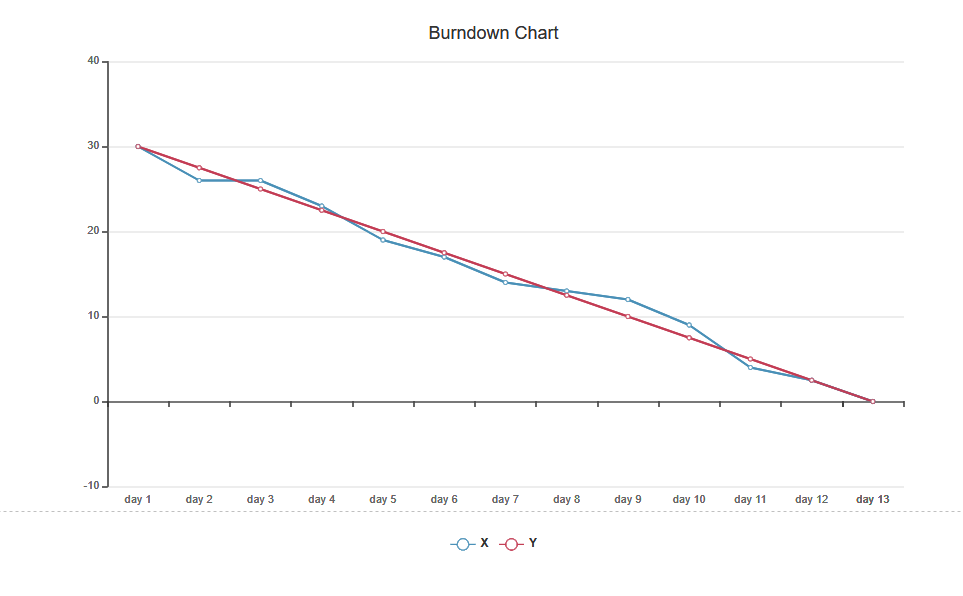


Figure 22 Burndown chart Sprint 1 Y as Story points of Sprint No.1 == 30 task points

# Bibliography

Atlassian, 2019. *Agile best practices and tutorials | Atlassian.* [Online]   
Available at: https://www.atlassian.com/agile  
[Accessed 01 April 2022].

docs.oracle.com, n.d. *Configure Role-Based Access Control (RBAC).* [Online]   
Available at: https://docs.oracle.com/cd/E65459\_01/admin.1112/e65449/content/general\_rbac.html  
[Accessed 01 April 2022].

Faller, P., 2019. *What Are User Personas and Why Are They Important? | Adobe XD Ideas.* [Online]   
Available at: https://xd.adobe.com/ideas/process/user-research/putting-personas-to-work-in-ux-design/  
[Accessed 01 April 2022].

Firebase, n.d. *Firebase A/B Testing | Optimize your app experience through experimentation.* [Online]   
Available at: https://firebase.google.com/products/ab-testing  
[Accessed 01 April 2022].

hibernate.org, n.d. *Your relational data. Objectively. - Hibernate ORM..* [Online]   
Available at: https://hibernate.org/orm/  
[Accessed 01 april 2022].

Oracle, 2019. *Abstract Methods and Classes (The JavaTM Tutorials > Learning the Java Language > Interfaces and Inheritance).* [Online]   
Available at: https://docs.oracle.com/javase/tutorial/java/IandI/abstract.html  
[Accessed 01 April 2022].

Paradigm, V., 2019. *UML Class Diagram Tutorial.* [Online]   
Available at: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/  
[Accessed 01 April 2022].

Scrum.org, 2019. *What is Scrum?.* [Online]   
Available at: https://www.scrum.org/resources/what-is-scrum  
[Accessed 01 April 2022].

Visual-paradigm.com, 2011. *How to Design Relational Database with ERD?.* [Online]   
Available at: https://www.visual-paradigm.com/tutorials/databasedesign.jsp  
[Accessed 01 April 2022].

Wikipedia, 2021. *High-level design.* [Online]   
Available at: https://en.wikipedia.org/wiki/High-level\_design  
[Accessed 01 April 2022].

www.bpmn.org, n.d. *BPMN Specification - Business Process Model and Notation.* [Online]   
Available at: https://www.bpmn.org/  
[Accessed 01 April 2022].