

Περιεχόμενα

Περιεχόμενα

1. ΕΙΣΑΓΩΓΗ	3
1.1. Σκοπός	3
1.2 Περίληψη	3
2. Τεχνολογία λογισμικού	4
2.1. Κύκλος ζωής λογισμικού.....	4
2.2 The information Systems Development Life Cycle (SDLC).	5
2.2.1 Αναλύοντας τον Κύκλο Ζωής Ανάπτυξης Συστήματος και τις Φάσεις του.	6
2.2.2 Μελέτη συστήματος - System study	6
2.2.3 Σχεδιασμός και ανάλυση Συστήματος.....	7
2.2.4 Διαχείριση με Cloud Computing	8
2.2.5. Υλοποίηση του λογισμικού εργαλείου	9
2.2.6. Γραφικό περιβάλλον	9
2.2.7. Συγγραφή Κώδικα	14
3. Ρόλος και μοντέλα ζωής λογισμικού	16
3.1. Προσδιορισμός των απαιτήσεων του έργου	17
4. Περιγραφή των καταστάσεων λειτουργίας – Περιπτώσεις χρήσης	23
5. Συμπεράσματα - εκτιμήσεις	25
6. Βιβλιογραφία	25

1. ΕΙΣΑΓΩΓΗ

Η πανδημία Covid19 έχει εξαπλωθεί με τον αριθμό των θυμάτων να αυξάνεται όλο και περισσότερο σε όλο τον κόσμο, έχοντας εξελιχθεί σε διεθνή κρίση δημόσιας υγείας. Οι κρατικοί φορείς σε πολλές χώρες έχουν μεριμνήσει για την ανάπτυξη εφαρμογών ανίχνευσης επαφών για την παρακολούθηση και τον εντοπισμό μολυσμένων ατόμων με σκοπό την μείωση της εξάπλωσης του ιού. Στρατηγικές ανίχνευσης όπως χειροκίνητος εντοπισμός επαφών και η συλλογή δεδομένων έχουν αναπτυχθεί για την διευκόλυνση των περιορισμών των lockdown.

Ωστόσο υπάρχουν πολλές ανησυχίες και μελέτες σχετικά με την αποτελεσματικότητα που προσφέρουν αυτές οι εφαρμογές, προβληματισμοί και δυσaráσκεια που δημιουργείται στον γενικό πληθυσμό λόγω της επέμβασης στις ατομικές ελευθερίες και της παραβίασης της ευρωπαϊκής νομοθεσίας περί προσωπικών δεδομένων.

1.1. Σκοπός

Ο σκοπός της εργασίας είναι να γίνει μελέτη για τη δημιουργία μίας εφαρμογής με το όνομα CovidPass, η οποία θα διευκολύνει το εθνικό σύστημα υγείας και τις αρχές επιβολής του νόμου στην ελεγχόμενη και ασφαλή διαχείριση της πανδημίας και της εξάπλωσης του ιού Covid19. Θα αξιοποιούνται τηρούμενες κρατικές βάσεις δεδομένων οι οποίες θα συγχρονίζονται και μέσω γραφικού περιβάλλοντος, θα ταυτοποιούνται μολυσμένα άτομα και οι επαφές τους.

Ειδικότερα θα ρυθμίζονται όλοι οι κλειστοί δημόσιοι χώροι, που θα είναι ελεύθερα προσιτοί προς το κοινό, οι οποίοι μπορεί να είναι εστιατόρια, καταστήματα, μουσεία, σχολεία, γραφεία, κινηματογράφοι, θέατρα, αεροδρόμια, αίθουσες αναμονής σταθμών λεωφορείων αλλά και δημόσια μέσα μαζικής μεταφοράς όπως πλοία, λεωφορεία, ταξί, αεροπλάνα, πορθμεία, τρένα ως «Δημόσιες Περιοχές».

Όταν ένα άτομο διαγνωστεί θετικό στον ιό Covid19, τα άτομα που έχουν μείνει στην ίδια ελεγχόμενη περιοχή εντός δύο τελευταίων ημερών και είχαν συνάντηση για περισσότερο από 15 λεπτά πρέπει να ενημερώνονται για αυτοαπομόνωση (καραντίνια) 14 ημερών και συνιστάται η πραγματοποίηση Covid19 test. Εκείνοι που είναι σε κατάσταση αυτοαπομόνωσης, δεν επιτρέπεται να εισέρχονται σε καμία δημόσια περιοχή πριν από την ολοκλήρωση του περιοριστικού μέτρου εκτός αν ελεγχθούν αρνητικά στον Covid19.

Το σύστημα θα αποθηκεύει την κατάσταση μόλυνσης covid19 του εισερχόμενου στο χώρο, η οποία μπορεί να έχει μία από τις ακόλουθες τρεις διαφορετικές καταστάσεις:

Πράσινη ένδειξη: Χωρίς μόλυνση Covid19. Θα αποτελούν τα άτομα που έχουν εξεταστεί με αρνητικό αποτέλεσμα στον Covid19.

Πορτοκαλί ένδειξη: Υποψία μόλυνσης Covid19. Θα αποτελούν τα άτομα που βρέθηκαν σε στενή επαφή με κρούσμα στις ελεγχόμενες περιοχές εντός δύο τελευταίων ημερών από το θετικό τεστ του κρούσματος.

Κόκκινη ένδειξη: Επιβεβαιωμένη μόλυνση από Covid19.

Κάθε άτομο δεν πρέπει να έχει περισσότερες από μία έγκυρες καταστάσεις CovidPass ανά πάσα στιγμή.

1.2 Περίληψη

Αρχικά περιγράφονται τα στάδια του κύκλου ζωής του λογισμικού, ορισμός του κύκλου ζωής ανάπτυξης των πληροφοριακών συστημάτων και η ανάπτυξη των επιμέρους διαδικασιών επί της εφαρμογής CovidPass που πρόκειται να υλοποιηθεί. Γίνεται ανάλυση του interface της εφαρμογής και περιγράφεται η διαδρομή λειτουργίας καθώς και η διαχείριση του μενού

χρήστη. Στην συνέχεια γίνεται αναφορά στην οργάνωση και υλοποίηση της εφαρμογής με σύγχρονες πλατφόρμες cloud που διαχειρίζονται Data Center. Ακολούθως περιγράφονται μέσω διαγραμμάτων dataflow chart & class diagram η συλλογή και διαχείριση πληροφοριών ανάμεσα στις βάσεις δεδομένων και της εφαρμογής καθώς και των κλάσεων που θα αναπτυχθούν για την υλοποίηση του έργου. Στην συνέχεια γίνεται σύντομη αναφορά στις ευέλικτες μεθόδους ανάπτυξης εφαρμογών και αναλύεται ο τρόπος διαχείρισης ανάπτυξης του λογισμικού CovidPass με τη μέθοδο Scrum. Γίνεται υποθετικός διαμοιρασμός ομάδων ανάπτυξης με συγκεκριμένους ρόλους. Περιγράφεται ο ρόλος του Scrum Master και Product Owner και παρουσιάζεται υποθετικό Product backlog και Sprint Backlog για τρεις κύκλους ανάπτυξης του έργου. Στην συνέχεια παρουσιάζονται περιπτώσεις χρήσης της εφαρμογής για τις τρεις καταστάσεις που θα υλοποιηθεί και παρουσιάζεται διάγραμμα Use Case για όλες τις περιπτώσεις χρήσης.

2. Τεχνολογία λογισμικού

Η εξελισσόμενη ψηφιακή επανάσταση, έχει ως αποτέλεσμα, την ανάπτυξη λογισμικού με ραγδαία εξέλιξη. Η ανάπτυξη λογισμικού δεν θα μπορούσε να παράγεται χωρίς την εφαρμογή μιας τυποποιημένης και συστηματικής προσέγγισης στην ανάλυση, σχεδίαση, υλοποίηση και συντήρηση με διαρκώς αυξανόμενο κύκλο εργασιών. Το λογισμικό πρέπει να υλοποιείται με συγκεκριμένες μεθοδολογίες, ώστε να διασφαλίζονται η μεγαλύτερη δυνατή ποιότητα και παραγωγικότητα και ταυτόχρονα το ελάχιστο δυνατό κόστος.

2.1. Κύκλος ζωής λογισμικού

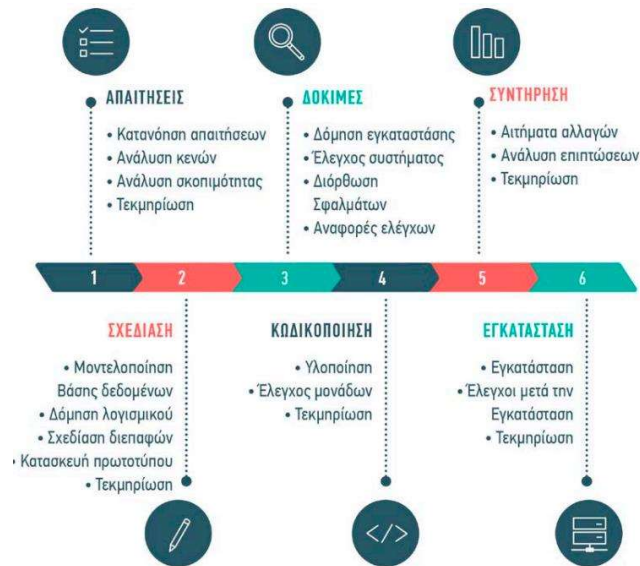
Αρχικά αναλύονται οι απαιτήσεις του συστήματος που θα αναπτυχθεί, ακολούθως συντάσσονται οι προδιαγραφές του λογισμικού και τέλος η ανάπτυξη και ο έλεγχος της λειτουργίας του. Πριν παραδοθεί για χρήση το λογισμικό απαιτείται να περάσει από κάποιες συγκεκριμένες δοκιμές (Testing).

Ως κύκλος ζωής λογισμικού (LifeCycle) ορίζεται το σύνολο των φάσεων από τις οποίες διέρχεται μια εφαρμογή λογισμικού, από τη σύλληψη της ιδέας, το σχεδιασμό, την ανάπτυξη, τον έλεγχο, τη λειτουργία, την αναβάθμιση, τη συντήρησή της, μέχρι και την απόσυρσή της. Τα στάδια ανάπτυξης λογισμικού στη γενική τους μορφή είναι τα εξής:

Απαιτήσεις: Περιλαμβάνει την καταγραφή των απαιτήσεων που πρέπει να πληροί το λογισμικό, οι οποίες διακρίνονται σε λειτουργικές και μη λειτουργικές.

Ανάλυση: Λαμβάνοντας υπόψη τις απαιτήσεις του συστήματος ακολουθεί η ανάλυση του λογισμικού με διάφορα μοντέλα, τα οποία το περιγράφουν διαγραμματικά τη δομή ανάπτυξης του λογισμικού. Τέτοια είναι τα flow charts, class diagrams, state charts κ.α.

Σχεδιασμός: Περιλαμβάνει το τελικό σχέδιο του λογισμικού, σύμφωνα με την προσέγγιση της δομημένης σχεδίασης. Στο στάδιο αυτό, καλύπτεται η αρχιτεκτονική σχεδίαση, η σχεδίαση διεπαφών με τον χρήστη, καθώς και η σχεδίαση της σχεσιακής βάσης δεδομένων που πιθανόν να χρειαστεί. Έτσι, ολοκληρώνεται το έργο του αναλυτή και ακολουθεί το έργο του προγραμματιστή.



Εικόνα1: Φάσεις ανάπτυξης λογισμικού

Πηγή: <https://repository.kallipos.gr/bitstream/11419/400/2>

2.2 The information Systems Development Life Cycle (SDLC).

Ο Κύκλος Ζωής Ανάπτυξης Πληροφοριακών Συστημάτων. Ο κύκλος ζωής του συστήματος είναι μια οργανωτική διαδικασία ανάπτυξης και συντήρησης συστημάτων. Βοηθά στη δημιουργία του έργου συστήματος, διότι παρέχει το σύνολο των διαδικασιών και των υποδιαδικασιών που απαιτούνται για την ανάπτυξη του συστήματος - λογισμικού. Ο κύκλος ζωής ανάπτυξης συστήματος (SDLC) είναι ένας επίσημος τρόπος διασφάλισης της εφαρμογής, επαρκών ελέγχων ασφαλείας και απαιτήσεων σε ένα νέο σύστημα - λογισμικό. Η ενσωμάτωση τεχνολογιών και πρακτικών στην ανάπτυξη νέων συστημάτων και εφαρμογών παρέχει μια ευκαιρία σχεδιασμού ασφάλειας σε συνδυασμό με το frontend που αποτελεί μέρος της διαδικασίας.

Ο κύκλος ζωής ανάπτυξης συστήματος σημαίνει κύκλο ζωής ανάπτυξης λογισμικού. Αποτελείται από τις ακόλουθες φάσεις:

- Μελέτη συστήματος - System study
- Μελέτη σκοπιμότητας - Feasibility study
- Ανάλυση συστήματος System analysis
- Σχεδιασμός συστήματος System design
- Κωδικοποίηση Coding
- Δοκιμές Testing
- Εφαρμογή Implementation
- Συντήρηση Maintenance

SDLC Phases

The entire SDLC process divided into the following SDLC steps:



SDLC Phases

Εικόνα 2: φάσεις Κύκλου Ζωής Ανάπτυξης Πληροφοριακών Συστημάτων.

Πηγή: <https://www.guru99.com/software-development-life-cycle-tutorial.html#4>

2.2.1 Αναλύοντας τον Κύκλο Ζωής Ανάπτυξης Συστήματος και τις Φάσεις του.

Αν και υπάρχουν πολυάριθμες εκδόσεις και ερμηνείες του Κύκλου Ζωής Ανάπτυξης Συστήματος, παρακάτω αναλύονται οι φάσεις και τα χαρακτηριστικά τους για την υλοποίηση της εφαρμογής CovidPass. Είναι σημαντικό να σημειωθεί ότι η διατήρηση ισχυρής επικοινωνίας με πελάτες – οργανισμούς & τελικούς χρήστες είναι ζωτικής σημασίας σε όλη τη διαδικασία.

2.2.2 Μελέτη συστήματος - System study

Η ιδέα ανάπτυξης του λογισμικού προέκυψε από την ανάγκη για την υποβοήθηση των αρχών επιβολής του νόμου καθώς και των υγειονομικών κανονισμών και πρωτοκόλλων για την προσπάθεια μείωσης εξάπλωσης του ιού covid19. Για την μέθοδο υλοποίησης του συστήματος θα γίνει χρήση Distributed Information System, διότι ο χρόνος υλοποίησης θα πρέπει να είναι πολύ σύντομος, εξαιτίας της υγειονομικής κρίσης που επικρατεί και της αναγκαιότητας περιορισμού των lockdown.

Πρώτη φάση:

Η εφαρμογή covidpass αποτελεί εφαρμογή σχεδιασμένη να λειτουργεί σε κινητές συσκευές για όλους τους τύπους λειτουργικών συστημάτων που είναι εγκατεστημένα σε αυτά (IOS, android). Δεδομένου ότι θα γίνεται χρήση από το σύνολο του πληθυσμού, η εφαρμογή πρέπει να είναι απλή και εύκολη στη χρήση, ακολουθούμενη από απλές και ευδιάκριτες οδηγίες χρήσης. Ο χρήστης θα κατεβάζει την εφαρμογή και θα την εγκαθιστά στη συσκευή του. Θα ταυτοποιείται ο χρήστης μέσω κυβερνητικής βάσης δεδομένων όπου τηρούνται τα φορολογικά στοιχεία του κάθε πολίτη, εξουσιοδοτώντας τη σχετική κρατική ιστοσελίδα φορολογικού μητρώου. Ακολουθώντας θα γίνεται ταυτοποίηση του προσώπου με αποστολή μηνύματος sms One Time Password (OTP) επιβεβαίωσης. Στην συνέχεια ο χρήστης θα κάνει checkin, όπου θα δηλώνει τον δημόσιο χώρο που πρόκειται να επισκεφτεί. Η ημερομηνία και η ώρα θα καταχωρούνται αυτόματα από το λειτουργικό σύστημα (sysdate). Όταν φεύγει από τον ελεγχόμενο δημόσιο χώρο, θα κάνει checkout και σε κεντρική βάση δεδομένων θα καταχωρούνται δεδομένα που αφορούν τον τόπο και την χρονική διάρκεια παραμονής του χρήστη στον κλειστό δημόσιο χώρο. Τέλος θα δημιουργηθεί βάση δεδομένων με την κατάσταση νόσησης του γενικού πληθυσμού, όπου θα καταγράφονται τα θετικά κρούσματα αλλά και τα στοιχεία των πολιτών που θα βρίσκονται στην πορτοκαλί ένδειξη.

Δεύτερη φάση

- 1 Downloading της εφαρμογής μόνο από πλατφόρμες appstore & googleplay (για λόγους ασφαλείας).

- 2 Είσοδος του χρήστη στην εφαρμογή με κωδικούς username & password με ανακατεύθυνση στην κεντρική σελίδα φορολογικού μητρώου.
- 3 Ταυτοποίηση του φυσικού προσώπου (χρήστη) μέσω ειδικού κωδικού (OTP) που θα του αποστέλλεται με sms στην τηλεφωνική σύνδεση που έχει καταχωρήσει σε προγενέστερο χρόνο στην ιστοσελίδα του φορολογικού μητρώου.
- 4 Επιλογή του κλειστού δημόσιου χώρου που πρόκειται να εισέλθει ως επισκέπτης.
- 5 Καταχώρηση σε εθνική βάση δεδομένων που θα δημιουργηθεί, όπου θα καταχωρούνται ο τόπος και ο χρόνος εισόδου του ατόμου σε συγκεκριμένο κλειστό χώρο της επιθυμίας του. Τα δεδομένα της ημερομηνίας και ώρας εισόδου θα αντλούνται αυτόματα από το λειτουργικό σύστημα.
- 6 Αυτόματη έκδοση QRcode όπου θα κωδικοποιούνται όλες οι παραπάνω πληροφορίες.
- 7 Συγκριτικός έλεγχος από υπεύθυνο κλειστού χώρου “σκανάροντας” το QRcode.
- 8 Έξοδος από την εφαρμογή, checkout με καταγραφή ημερομηνίας και ώρας και καταχώρηση σε κρατική βάση δεδομένων.

Προβλήματα που μπορεί να παρουσιαστούν:

Πιθανό πρόβλημα ως προς το πεδίο εφαρμογής είναι η μη χρήση συσκευής smartphone από άτομα που είτε δεν έχουν είτε δεν δύναται να χρησιμοποιήσουν. Επίσης ο τρόπος εφαρμογής και η καταγραφή των ατόμων μπορεί δημιουργήσει προβληματισμούς στο κοινωνικό σύνολο όσον αφορά τον ευρωπαϊκό κανονισμό για τη προστασία προσωπικών δεδομένων.

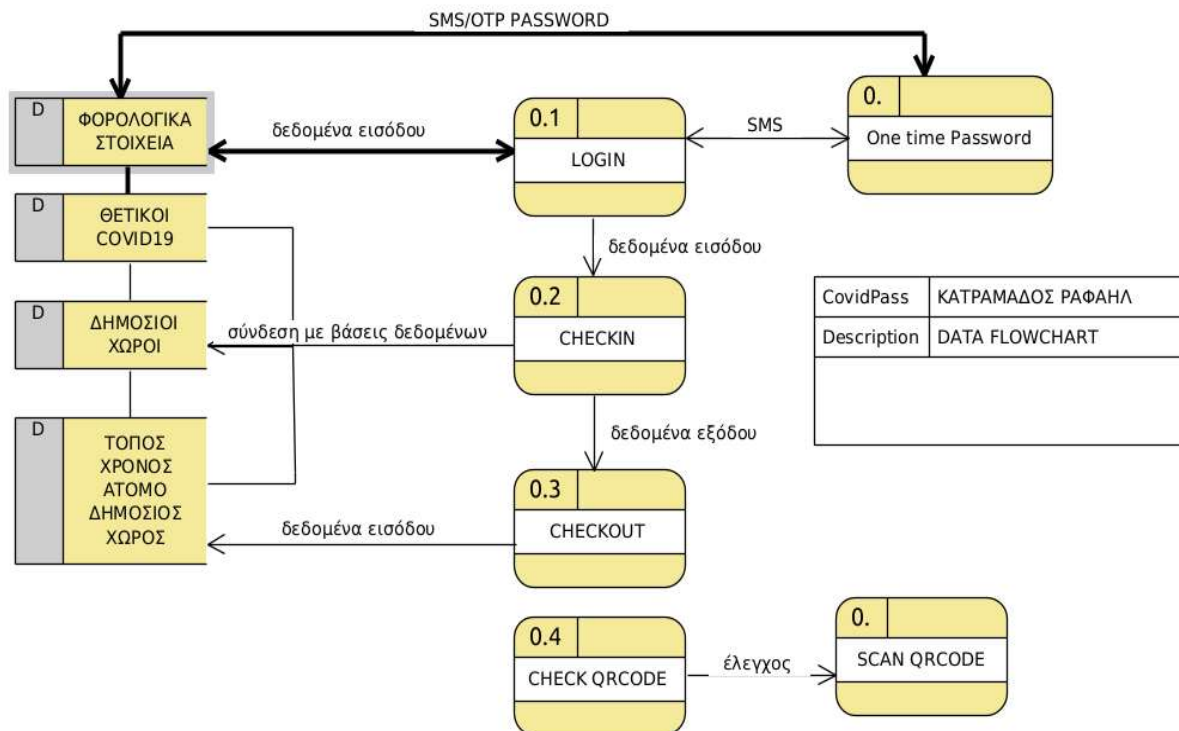
2.2.3 Σχεδιασμός και ανάλυση Συστήματος

Ο σχεδιασμός της εφαρμογής επικεντρώθηκε σε βασικά θέματα με σκοπό τη λήψη αποφάσεων σχετικών με τη μετέπειτα διευκόλυνση του επόμενου σταδίου της υλοποίησης, δηλαδή της συγγραφής πηγαίου κώδικα. Ως γλώσσα προγραμματισμού μπορεί να καθοριστεί η java με δεδομένο ότι η εφαρμογή προορίζεται να λειτουργεί σε περιβάλλον κινητών συσκευών και ότι κρίνεται πιο κατάλληλη γλώσσα για τις ανάγκες υλοποίησης από άποψη δυνατοτήτων και διαδικτυακής χρησιμότητας της εφαρμογής. Έπειτα θα επιλεγεί ως περιβάλλον ανάπτυξης (IDE), το Microsoft Visual Studio Code, καθώς είναι ένα πλήρες εργαλείο υλοποίησης κώδικα. Επίσης, για λόγους συμβατότητας στην διαχείριση των βάσεων δεδομένων θα επιλεγεί η mysql με εφαρμογή του εργαλείου SQL Server Management Studio.

Εν συνεχεία, θα ακολουθηθεί ο γενικός αρχιτεκτονικός σχεδιασμός με γνώμονα την εργονομικότητα της γραφικής διεπαφής χρήστη (GUI), την ευκολία της χρήσης, την απλότητα παρουσίασης των δεδομένων και την ελαχιστοποίηση των απαιτούμενων οδηγιών χρήσης.

Ακολουθώντας τις προδιαγραφές, θεωρήθηκε ότι αυτές εξυπηρετούνται αν σε ένα ενιαίο παράθυρο (window) του εργαλείου επιμεριστούν με βασικά και εμφανή κουμπιά οι λειτουργίες της εφαρμογής. Έτσι, αναλογεί ένα button για την σύνδεση του χρήστη με βάση δεδομένων του φορολογικού μητρώου για την ταυτοποίησή του (login). Δεύτερο button (checkin) για τη σύνδεση του χρήστη με βάση δεδομένων όπου θα καταχωρούνται όλοι οι δημόσιοι κλειστοί χώροι και ο χρήστης θα κάνει την αντίστοιχη επιλογή. Τρίτο button για τη σύνδεση του χρήστη με βάση δεδομένων προκειμένου να γίνει έξοδος από το σύστημα (checkout) με καταχώρηση του χρόνου εξόδου από τον δημόσιο χώρο. Τέταρτο button εμφάνιση scanner Quick Response code (QRcode) όπου θα ελέγχεται ο χρήστης και θα εμφανίζεται η ανάλογη κατάσταση. Ο έλεγχος θα γίνεται αποκωδικοποιώντας όλα τα δεδομένα και με βάση την ημερομηνία του συστήματος και ενός αλγορίθμου θα εμφανίζεται η αντίστοιχη κατάσταση. Τέλος απαραίτητη θεωρήθηκε η ύπαρξη ενός πέμπτου button (help), με πληροφορίες για το χρήστη σχετικά με την εφαρμογή και με εμφάνιση χρήσιμου τηλεφωνικού αριθμού για τεχνική υποστήριξη και περεταίρω πληροφορίες.

Το στάδιο του σχεδιασμού θα ολοκληρωθεί με το διάγραμμα ροής δεδομένων (dataflow chart), τον καθορισμό των δεδομένων εισόδου και δεδομένων εξόδου και την επεξεργασία αυτών (διάγραμμα 1).



(ΔΙΑΓΡΑΜΜΑ 1: DATAFLOW CHART)

2.2.4 Διαχείριση με Cloud Computing

Όσον αφορά τη δικτύωση, τη διαχείριση των βάσεων δεδομένων και τα ζητήματα της αποθήκευσης και γενικότερα τους πόρους που θα χρησιμοποιηθούν, προτείνεται η διαχείρισή τους με cloud computing. Η εφαρμογή CovidPass θα εξυπηρετεί ταυτόχρονα πολλούς χρήστες. Ο αριθμός των χρηστών όμως μπορεί να είναι απρόβλεπτος, με αποτέλεσμα να υπάρχει πιθανότητα πολλών ταυτόχρονων checkin σε συγκεκριμένες ώρες και ημέρες, με συνέπεια την απότομη αύξηση της κίνησης του δικτύου. Πχ μπορεί να γίνουν ταυτόχρονα Checkin από εκατοντάδες χρήστες που θα παρακολουθήσουν την ίδια θεατρική παράσταση. Η χωρητικότητα του δικτύου θα πρέπει να επαρκεί, ώστε να εξυπηρετεί με την ταχύτητα που θα χρειαστεί και τον αριθμό των ταυτόχρονων χρηστών που θα συνδεθούν πάνω στην ώρα αιχμής.

Η εφαρμογή θα έχει ως χαρακτηριστικό την αυτοεξυπηρέτηση (on – demand – self – service). Θα υπάρχει ευρεία πρόσβαση στο δίκτυο. Θα υπάρξει καλύτερη οργάνωση με τη διάθεση πόρων που παρέχουν δυνατότητα μέτρησης χρησιμοποιούμενων υπηρεσιών (πχ διαθέσιμα user accounts, χώροι αποθήκευσης).

Με τα Virtual Machines σε cloud (πχ με χρήση Microsoft Azure) μπορεί να γίνει διαχείριση της εφαρμογής με μεγάλες αυξομειώσεις σε απαιτήσεις και σε υπολογιστικούς πόρους και upgrades των διαθέσιμων πόρων χωρίς να χαθεί πολύτιμος χρόνος. Τέτοιες πλατφόρμες μπορούν χρησιμοποιηθούν, για να γίνει διαχείριση ανάλογων εφαρμογών όπως SQL server.

Ως αρχιτεκτονική μπορεί να οριστεί αυτή του Hybrid Cloud. Για την δημιουργία του απαιτείται η ύπαρξη ενός private cloud που θα υφίσταται εντός του οργανισμού που θα υλοποιήσει την εφαρμογή CovidPass ή κάποιου ιδιωτικού φορέα παροχής υπηρεσιών cloud. Η σύνδεση δύναται να πραγματοποιηθεί με Public υποδομή ως υπηρεσία (IaaS) όπως πχ το

AWS Services ή Microsoft Azure. Απαραίτητη προϋπόθεση είναι η συνδεσιμότητα δικτύου WAN μεταξύ αυτών των δύο δομών.

Ασφάλεια στο cloud

Επίσης θα πρέπει να γίνει μερίμνα για εφαρμογή Region Pair και συνδεσιμότητα με Availability zone, δηλαδή συνδεσιμότητα του Storage και όλων των διαθέσιμων πόρων της εφαρμογής με ένα δεύτερο data center, έτσι ώστε να αποφευχθεί η απώλεια δεδομένων αλλά και η μη λειτουργία της εφαρμογής. Με αυτόν τον τρόπο θα γίνει mirroring και η υπηρεσία θα είναι πάντα διαθέσιμη χωρίς να σταματήσει ποτέ η λειτουργία της, σε περίπτωση που θα γίνει κάποιο uptime λόγω κάποιας διακοπής ρεύματος ή φυσικής καταστροφής ή κάποιας διαδικτυακής επίθεσης. Με αυτόν τον τρόπο διασφαλίζεται το redundancy, το recovery και το uptime της εφαρμογής.

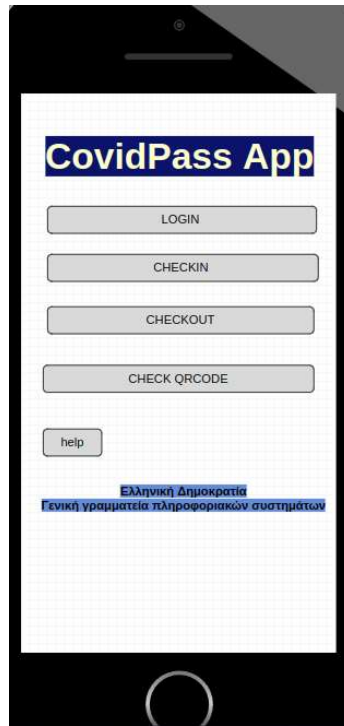
2.2.5. Υλοποίηση του λογισμικού εργαλείου

Η υλοποίηση, δηλαδή η γραφή κώδικα στη γλώσσα προγραμματισμού Java, ακολουθεί τον αρχικό σχεδιασμό της εφαρμογής με τη μέθοδο code and fix. Στόχος είναι η ελαχιστοποίηση της πολυπλοκότητας και η πρόβλεψη αλλαγών και επεκτάσεων με την κατάλληλη δομή του κώδικα, αλλά και τον επαρκή σχολιασμό του, ώστε να διευκολύνεται η κατανόησή του από τρίτους και κατ' επέκταση η συμπλήρωση, η διόρθωση και η επαναχρησιμοποίησή του.

2.2.6. Γραφικό περιβάλλον

Αρχικά σχεδιάζονται οι οθόνες της εφαρμογής με όλες τις βασικές λειτουργίες υλοποίησης, προκειμένου να πραγματοποιηθεί η διεπαφή χρήστη (GUI) βάσει του σχεδιασμού, αποτελώντας μια επαναληπτική διαδικασία. Πρέπει να ληφθεί υπόψη ότι η είσοδος στους κλειστούς δημόσιους χώρους θα πραγματοποιείται από το γενικό σύνολο του πληθυσμού και η χρήση θα προέρχεται από ένα ευρύ φάσμα ηλικιών κάθε μορφωτικού επιπέδου (φοιτητές μαθητές, συνταξιούχους) και για αυτό το λόγο πρέπει να είναι εύκολη και φιλική στο χειρισμό. Έτσι θα σχεδιαστούν πέντε buttons σε ένα παράθυρο, κάθε ένα από τα οποία αντιστοιχεί στα επιμέρους στοιχεία του υπό κατασκευή λογισμικού, όπως αρχικά είχε προβλεφθεί: Button για την σύνδεση και ανακατατεύθυνση του χρήστη σε βάση δεδομένων του φορολογικού μητρώου για την ταυτοποίησή του (login) και την επαλήθευση με ταυτοπροσωπία με OTP (One Time Password). Δεύτερο button (checkin) για τη σύνδεση του χρήστη με βάση δεδομένων όπου θα καταχωρούνται όλοι οι δημόσιοι κλειστοί χώροι και ο χρήστης θα κάνει την αντίστοιχη επιλογή εισόδου. Τρίτο button για τη σύνδεση του χρήστη με βάση δεδομένων προκειμένου να γίνει έξοδος από την εφαρμογή (checkout) με καταχώρηση του χρόνου εξόδου από τον δημόσιο χώρο όπου είχε εισέλθει. Τέταρτο button εμφάνιση scanner Quick Response code (QRcode) με τον οποίο θα ελέγχεται ο χρήστης από υπεύθυνο που έχει ορισθεί στο δημόσιο κλειστό χώρο. Ο έλεγχος θα γίνεται αποκωδικοποιώντας όλα τα δεδομένα και με βάση την ημερομηνία του συστήματος και ενός αλγορίθμου θα εμφανίζεται η αντίστοιχη κατάσταση. Τέλος πέμπτο button help με πληροφορίες σχετικά με την εφαρμογή.

Στις παρακάτω εικόνες παρουσιάζονται οι οθόνες με το interface του χρήστη στην εφαρμογή CovidPass, βήμα προς βήμα ως προς τις επιλογές που θα πραγματοποιήσει ο χρήστης. Στην εικόνα 3 που ακολουθεί παρουσιάζεται το αρχικό interface της εφαρμογής, όπως θα εμφανίζεται στον χρήστη.



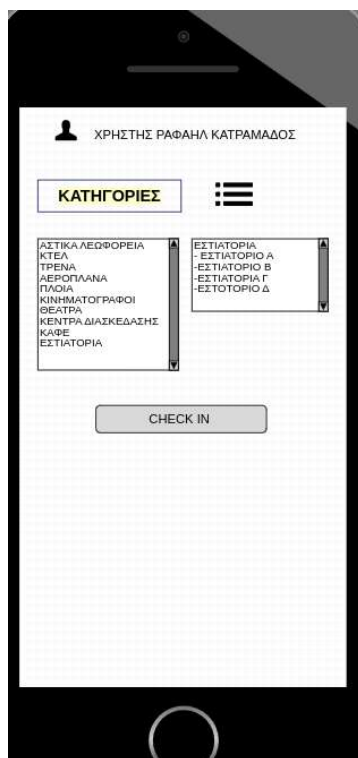
(εικόνα 3: interface χρήστη – αρχική εμφάνιση του βασικού μενού)

Πρώτο βήμα είναι ο χρήστης να κάνει είσοδο (login) στην εφαρμογή πληκτρολογώντας username & password προκειμένου να γίνει η ταυτοποίησή του από το σύστημα. Αμέσως μετά θα γίνεται ανακατεύθυνση στην σελίδα φορολογικού μητρώου όπου θα γίνεται η ταυτοποίηση του χρήστη (εικόνα 4).



(Εικόνα 4: Interface χρήστη login με ingredients φορολογικού μητρώου)

Αφού γίνει με επιτυχία η ταυτοποίηση του χρήστη, το δεύτερο βήμα είναι να επιλέξει το button CHECKIN για την επιλογή του δημόσιου χώρου που πρόκειται να εισέλθει μέσω μιας λίστας με κατηγορίες και υποκατηγορίες κλειστών δημοσίων χώρων. Επιπροσθέτως στο πάνω μέρος της οθόνης θα εμφανίζεται το όνομα και το επίθετο του χρήστη που έκανε login στην εφαρμογή. Στην εικόνα 5 που ακολουθεί εμφανίζεται το interface με στοιχεία υποθετικού χρήστη που έκανε login και τις κατηγορίες και υποκατηγορίες των κλειστών δημοσίων χώρων που θα εμφανίζονται προς επιλογή του χρήστη.



(Εικόνα 5 Γραφικό περιβάλλον CHECKIN)

Αφού έχουν ολοκληρωθεί τα προηγούμενα βήματα με επιτυχία, θα επιστρέφεται στον χρήστη QRcode με κωδικοποιημένη την κατάσταση νόσησής του. Στο πάνω μέρος του παραθύρου θα εμφανίζεται το όνομα και το επίθετο του χρήστη που έχει κάνει CHECKIN και το όνομα του κλειστού δημόσιου χώρου που επιθυμεί να εισέλθει. (εικόνα 6).



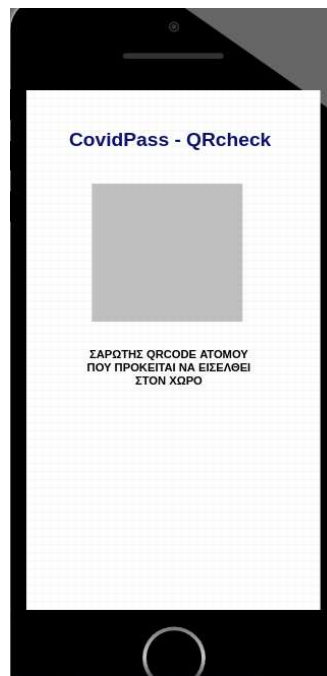
(Εικόνα 6: Interface έκδοση QR CODE χρήση)

Στην συνέχεια στην εικόνα 7 παρουσιάζεται το γραφικό περιβάλλον που αφορά την έξοδο του χρήστη από τον δημόσιο χώρο. Θα γίνεται επιλογή του button CHECKOUT όταν ο χρήστης αποχωρεί από τον χώρο που παρευρίσκεται και θα επιβεβαιώνεται η έξοδος από το σύστημα με σχετικό μήνυμα alert επιβεβαίωσης απάντησης.



(Εικόνα: 7 interface CHECKOUT)

Επιπροσθέτως η εφαρμογή θα έχει την επιλογή – button CHECK QRCODE, όπου ο υπεύθυνος του ανάλογου δημόσιου κλειστού χώρου θα σκανάρει το QRcode του χρήστη που πρόκειται να εισέλθει και μέσω αλγορίθμου και με αυτό τον τρόπο θα εμφανίζεται η αντίστοιχη κατάσταση νόσησης. Στην εικόνα 8 που ακολουθεί εμφανίζεται το interface CHECK QRCODE.



(Εικόνα 8: Περιβάλλον CHECK QRCODE)

Στις παρακάτω εικόνες 9,10 και 11 εμφανίζεται το output του ελέγχου, για τις τρεις καταστάσεις νόσησης σε μορφή Quick Response Code.



Εικόνα 9

Χωρίς μόλυνση Covid19

Πράσινη ένδειξη



Εικόνα 10

Επιβεβαιωμένη μόλυνση

κόκκινη ένδειξη



Εικόνα 11

Πιθανή Μόλυνση

Πορτοκαλί ένδειξη



Τέλος ο κάθε χρήστης θα έχει δυνατότητα επιλογής του button help, όπου θα εμφανίζονται πληροφορίες για τη χρήση της εφαρμογής καθώς και τηλεφωνικός αριθμός επικοινωνίας για τυχόν τηλεφωνική υποστήριξη (εικόνα 12).



(Εικόνα 12: Button help)

Για την περίπτωση που κάποιο άτομο είχε εισέλθει σε κλειστό δημόσιο χώρο και έχει διαγνωσθεί μεταγενέστερα θετικό στον ιό, τότε όλα τα άλλα άτομα που είχαν μείνει στην ίδια ελεγχόμενη περιοχή εντός δύο ημερών και είχαν επαφή για περισσότερο από 15 λεπτά, θα ενημερώνονται με αυτοματοποιημένο μήνυμα στον αριθμό της τηλεφωνικής σύνδεσης που είχαν κάνει είσοδο (login), CHECKIN και είχαν ταυτοποιηθεί ως χρήστες. Το μήνυμα θα αναγράφει: *"Ανίχνευση θετικού κρούσματος την ηη/μμ/εε στην περιοχή "area", παρακαλείσθε όπως προβείτε σε διαγνωστικό έλεγχο ή αυτοαπομόνωση δεκατεσσάρων ημερών"*. Ακολούθως θα ενημερώνεται η βάση δεδομένων νόσησης με το αντίστοιχο αποτέλεσμα και η κατάσταση της εφαρμογής θα μετατρέπεται σε πορτοκαλί, *"πιθανή μόλυνση από Covid19"*.

Εάν πραγματοποιηθεί αρνητικός διαγνωστικός έλεγχος, τότε θα ενημερώνεται η βάση δεδομένων νόσησης αυτοματοποιημένα και η κατάσταση της εφαρμογής θα μετατρέπεται σε πράσινη *"Χωρίς Covid19"*.

Εάν πραγματοποιηθεί θετικός διαγνωστικός έλεγχος τότε θα ενημερώνεται η βάση δεδομένων νόσησης αυτοματοποιημένα και η κατάσταση της εφαρμογής θα μετατρέπεται σε κόκκινη *"επιβεβαιωμένο κρούσμα Covid19"*.

Κατά τον διαγνωστικό έλεγχο, το ελεγχόμενο άτομο θα κοινοποιεί στον υγειονομικό υπάλληλο, τον Α.Φ.Μ. του και τον αριθμό της τηλεφωνικής σύνδεσης για την οποία είναι χρήστης. Έτσι το σύστημα της βάσης δεδομένων φορολογικού μητρώου θα συγχρονίζεται αυτόματα με το αποτέλεσμα του ελέγχου και θα αλλάζει ανάλογα η κατάσταση νόσησης.

Η διαδικασία αυτή θα πραγματοποιείται με συγχρονισμό της βάσης δεδομένων νόσησης με αυτή της καταγραφής των ατόμων των κλειστών δημόσιων χώρων. Θα γίνεται έλεγχος μέσω αλγορίθμου που θα εντοπίζει τα άτομα που είχαν μείνει στην ίδια ελεγχόμενη περιοχή κατά το χρονικό διάστημα που προαναφέρθηκε μέσω προγραμματιστικών ελέγχων, αλγορίθμων και συνθηκών. Στην περίπτωση αληθών συνθηκών θα αποστέλλεται SMS με το σχετικό μήνυμα στον αριθμό τηλεφωνικής σύνδεσης των εμπλεκόμενων ατόμων.

2.2.7. Συγγραφή Κώδικα

Στη συνέχεια αναφέρεται το στάδιο της δημιουργίας πηγαίου κώδικα. Ο κώδικας δομήθηκε σε γενικές ενότητες (regions) σε αντιστοιχία των απαιτούμενων σταδίων βάσει του σχεδιασμού. Επιγραμματικά, αναφέρονται οι ενότητες αυτές όπως θα εμφανίζονται στην εφαρμογή και αναφέρονται οι κύριες κλάσεις που αποτελούν και τον κορμό του προγράμματος, χωρίς τις συντακτικές τους λεπτομέρειες. Ακολούθως παρουσιάζεται διάγραμμα κλάσεων όπου απεικονίζονται οι κλάσεις που θα χρησιμοποιηθούν και οι συσχετίσεις μεταξύ τους (διάγραμμα 2).

- **Κλάση CovidPass**

Αντιπροσωπεύει την εφαρμογή.

Λειτουργίες της κλάσης:

Disconnect: Γίνεται αποσύνδεση του χρήστη από την εφαρμογή.

Connect: Σύνδεση του χρήστη με την εφαρμογή.

CameraSwitch: Άνοιγμα – λειτουργία κάμερας της συσκευής για Scanning QR CODE.

Mobile_Authentication: Αποστολή και επιβεβαίωση (OTP).

- **Κλάση LoginSend**

getUserData(): Αντλεί τα δεδομένα που αφορούν τον χρήστη της συσκευής.

getmobilenumber: Αντλεί τον αριθμό τηλεφωνικής σύνδεσης του χρήστη και την επαλήθεύει.

getHealthStatus: Συγχρονισμός με βάση δεδομένων σχετικά με την κατάσταση υγείας του χρήστη.

getDecodeImage() : Γίνεται κωδικοποίηση όλων των δεδομένων σε εικόνα QR CODE

- **Κλάση AddExitData**

GetPlaceName: Αντλείται και καταχωρείται το όνομα και το μέρος των κλειστών δημόσιων χώρων.

GetDateTime: Αντλείται και καταχωρείται η ημέρα και η ώρα εισόδου και εξόδου.

- **Κλάση QRcodeRetriever**

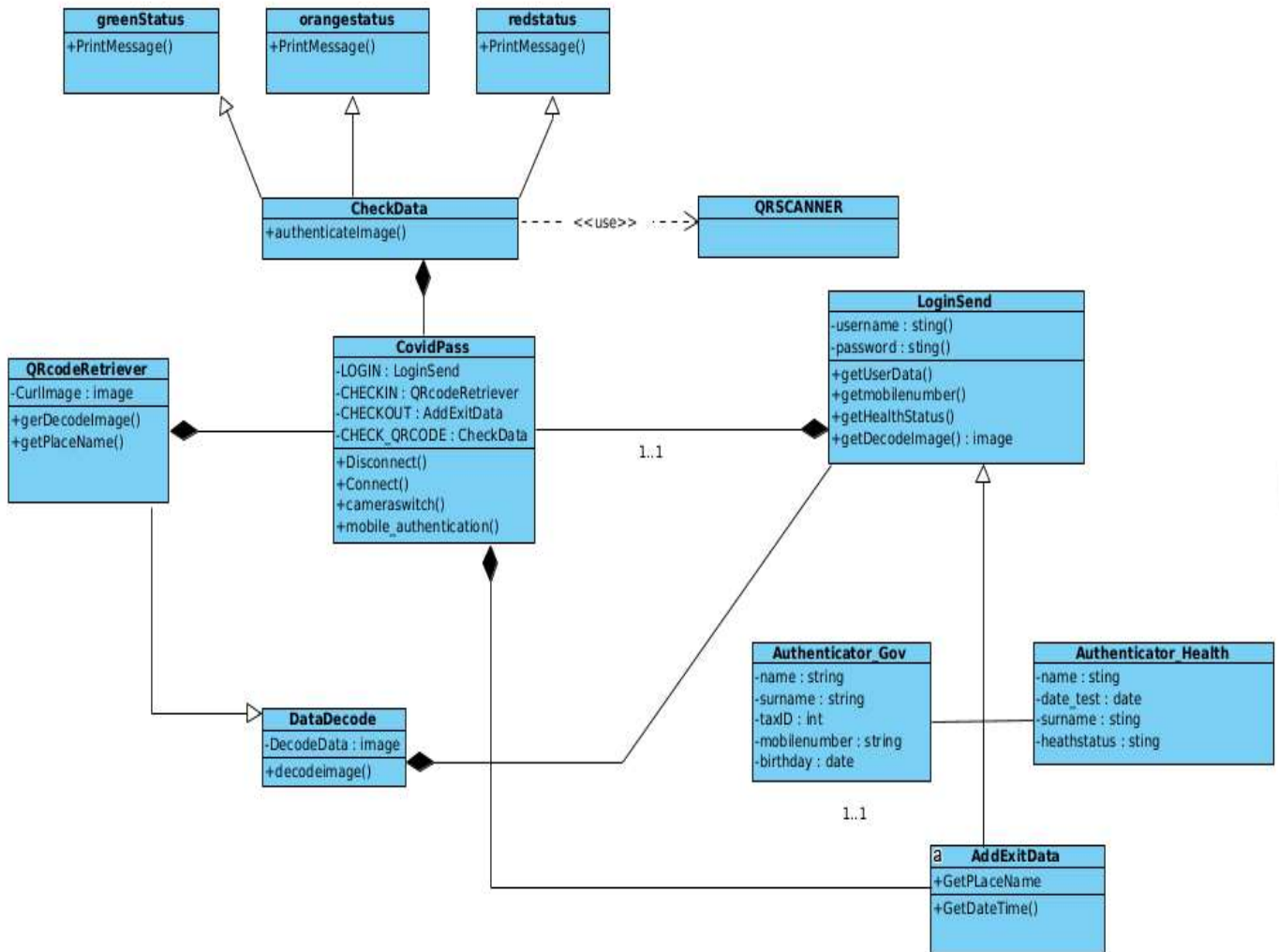
Συνδέεται από την κλάση DataDecode με λειτουργίες της gerDecodeImage & getPlaceName με σκοπό την κωδικοποίηση και την εμφάνιση του QRcode με όλες τις πληροφορίες καθώς και το όνομα του κλειστού χώρου.

- **Κλάση CheckData**

Αποτελείται από τις κλάσεις greenstatus, orangetatus, redstatus καθώς και από την QRSCANNER με σκοπό την άντληση και εμφάνιση πληροφοριών σχετικά με την κατάσταση που βρίσκεται ο χρήστης.

- **Κλάσεις Authenticator Gov, Authenticator Health & AddExitData**

Αποτελούν κλάσεις που τηρούν και διαμοιράζονται δεδομένα των βάσεων που συγχρονίζονται και αφορούν στοιχεία φορολογικού μητρώου του χρήστη, την υγειονομική κατάσταση που αφορά τον covid19 καθώς και τα μέρη που αποτελούν δημόσιοι χώροι.



(ΔΙΑΓΡΑΜΜΑ 2: Το διάγραμμα κλάσεων CovidPass)

2.2.8. Έλεγχος και διόρθωση σφαλμάτων

Η φάση του ελέγχου και των δοκιμών θα είναι ενσωματωμένη στους επαναληπτικούς κύκλους της φάσης υλοποίησης, μέσω της πρακτικής «δοκιμής και λάθους» (trial and error). Μετά την προσθήκη του κάθε επιπλέον στοιχείου στον κώδικα, θα πραγματοποιείται έλεγχος σε πραγματικές συνθήκες και ακολουθούσε η διαδικασία της αποσφαλμάτωσης (debugging). Επιπροσθέτως, έμφαση θα δοθεί στην πρόβλεψη λανθασμένης σειράς ενεργειών από μεριάς του τελικού χρήστη. Αν για παράδειγμα δεν ακολουθηθεί με τη σειρά η ολοκλήρωση των λειτουργιών κάθε button και ένας χρήστης μεταβεί κατευθείαν στην επιλογή CHECKIN, χωρίς να κάνει αρχικά την επιλογή LOGIN.

2.2.9. Συντήρηση

Η διαδικασία της συντήρησης και βελτίωσης του εργαλείου έχει να κάνει με τη μελλοντική χρήση του. Στις πλατφόρμες από όπου διατίθεται προς downloading, θα προτρέπονται οι χρήστες να σχολιάσουν και να υποδείξουν βελτιώσεις και σφάλματα με σχόλια. Τα σχόλια των χρηστών θα λειτουργήσουν ως ανάδραση για την περαιτέρω βελτίωση και επέκταση των δυνατοτήτων της εφαρμογής, γεγονός το οποίο με τη σειρά του θα οδηγήσει στον μελλοντικό εμπλουτισμό της εφαρμογής.

3. Ρόλος και μοντέλα κύκλου ζωής λογισμικού

Ρόλος του μοντέλου κύκλου ζωής λογισμικού είναι να καθοδηγήσει τον μηχανικό λογισμικού στις ενέργειες που απαιτούνται, ώστε να επιτευχθεί το καλύτερο δυνατό αποτέλεσμα σε όσο

το δυνατόν καλύτερη υλοποίηση και παραγωγικότερη ανάπτυξη των βημάτων του λογισμικού, με τα λιγότερα δυνατά σφάλματα, το μικρότερο δυνατό ρίσκο (οικονομικό & χρονικό) και το χαμηλότερο κόστος. Οι παραπάνω παράγοντες μπορεί να διαφοροποιούνται ανάλογα με το μέγεθος και το πεδίο εφαρμογής του λογισμικού. Τα συνηθέστερα μοντέλα κύκλου ζωής λογισμικού είναι:

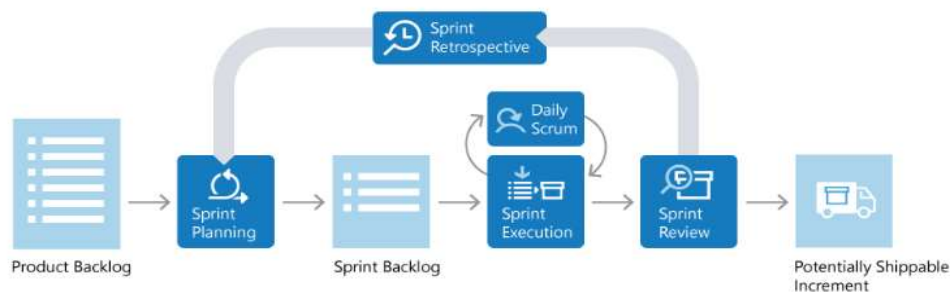
1. Το Μοντέλο του Καταρράκτη.
2. Το Μοντέλο Πρωτοτυποποίησης.
3. Το Μοντέλο της Λειτουργικής Επαύξησης
4. Το Σπειροειδές Μοντέλο
5. Το Μοντέλο του Πίδακα
6. Το Ευέλικτο Μοντέλο (agile)
7. Scrum

3.1. Προσδιορισμός των απαιτήσεων του έργου

Υλοποίηση με μεθοδολογία Scrum

Αποσκοπώντας στην επίτευξη της καλύτερης οργάνωσης και δραστηριότητας θα ακολουθηθεί η μεθοδολογία Scrum. Αποτελεί μια ευέλικτη πρακτική, που στοχεύει στην ανάπτυξη και διαχείριση του λογισμικού. Αποτελεί μια καλή πρακτική για ομάδες εργασίας με μεγάλο όγκο δουλειάς και υψηλές απαιτήσεις. Η ανάπτυξη του λογισμικού με αυτή τη μεθοδολογία θα αναλυθεί από πολλές και διαφορετικές οπτικές με βάση τους ανθρώπους που θα απασχοληθούν και την διαχείριση του. Υπάρχει σωστή διαχείριση του χρόνου εργασίας. Καθορίζει μια ευέλικτη, ολιστική μέθοδο για την ανάπτυξη του έργου, όπου μια ομάδα ανάπτυξης λειτουργεί με στόχο την κοινή επίτευξη του τελικού σκοπού. Η ανάπτυξη γίνεται σε επαναληπτικούς κύκλους εργασίας, οι οποίοι ονομάζονται Sprints. Οι επαναλήψεις δεν πρέπει να διαρκούν περισσότερο από ένα μήνα και εκτελούνται διαδοχικά χωρίς χρονικές καθυστερήσεις μεταξύ τους. Τα Sprints έχουν συγκεκριμένο χρονοδιάγραμμα (timeline), το οποίο ποτέ δεν παρακάμπτεται, ακόμη και αν η καθορισμένη εργασία δεν έχει ολοκληρωθεί. Στην αρχή κάθε Sprint η ομάδα επιλέγει τα στοιχεία, τα οποία αποτελούν τις απαιτήσεις του οργανισμού, από μια λίστα στην οποία είναι ταξινομημένες βάσει προτεραιότητας (Backlog). Τα στοιχεία αυτά δεν μπορούν να αλλάξουν κατά τη διάρκεια του Sprint. Κάθε μέρα η ομάδα οργανώνει σύντομες συγκεντρώσεις για να ελέγξει την πρόοδό της και για να προσδιοριστούν τα επόμενα βήματα που χρειάζονται για την ολοκλήρωση της εργασίας.

Στο τέλος του Sprint η ομάδα παρουσιάζει το αποτέλεσμα στον πελάτη και συγκεντρώνει σχόλια, διορθώσεις και παραλήψεις που θα ενσωματωθούν στο επόμενο Sprint.



Εικόνα 13 The Scrum Lifecycle

(πηγή: <https://docs.microsoft.com/en-us/devops/plan/what-is-scrum>)

3.2.1 Ρόλοι και ευθύνες

The Developer team (Η ομάδα ανάπτυξης): Για την ολοκλήρωση του έργου θα συνεργαστούν τρεις διαφορετικές ομάδες των οκτώ ατόμων. Ένας product owner ορίζεται ως υπεύθυ-

νος και για τις τρεις ομάδες. Κάθε ομάδα θα απαρτίζεται από τον δικό της scrum master, από τρία άτομα που θα αποτελούν programming developers (προγραμματιστές λογισμικού), δύο που θα πραγματοποιούν δοκιμές του λογισμικού και δύο μέλη της ομάδας που θα καταγράφουν τις απαιτήσεις που προκύπτουν. Όλες οι ομάδες θα απασχολούν αναλυτές λογισμικού, testers ανάπτυξης και δοκιμής του λογισμικού και σχεδιαστές βάσεων δεδομένων (DB Designers).

Το υλικό και ο κώδικας που θα χρησιμοποιηθούν θα πρέπει να ελέγχονται, να συντηρούνται και να ανανεώνονται συνεχώς από την ομάδα ανάπτυξης λογισμικού μελετώντας τυχόν σφάλματα που θα προκύψουν κατά την εκτέλεσή τους. Επιπροσθέτως θα πρέπει να γίνεται συντήρηση των τηρούμενων βάσεων δεδομένων που αναφέρθηκαν.

Ειδικότερα, η πρώτη ομάδα θα απασχοληθεί με το back-end της εφαρμογής, δηλαδή τον σχεδιασμό των βάσεων δεδομένων, τον συγχρονισμό, τον έλεγχο των καταχωρημένων στοιχείων που αφορούν την καταγραφή των δημόσιων χώρων, το συγχρονισμό με τη βάση του φορολογικού μητρώου των πολιτών και την υλοποίηση μέρους του λογισμικού που αφορά την κατάσταση νόσησης. Επιπροσθέτως θα φροντίσει για την επιλογή και συγγραφή του μεγαλύτερου μέρους του κώδικα και την τυχόν ανανέωση και συντήρησή του. Θα πραγματοποιεί δοκιμές της υλοποίησης και διόρθωση των σφαλμάτων που τυχόν θα προκύψουν. Επιπλέον θα πραγματοποιηθεί επέκταση λογισμικού η οποία θα συγχρονίζει τη βάση δεδομένων φορολογικού μητρώου πολιτών με το χρήστη της τηλεφωνικής σύνδεσης για την οποία είναι κάτοχος. Σε περίπτωση χρήσης cloud computing που προαναφέρθηκε, θα μεριμνήσει για την διαχείριση των απαραίτητων πόρων που θα χρειαστούν, την διαχείριση της ανάλογης πλατφόρμας καθώς και την απαραίτητη δικτύωση που θα χρειαστεί. Επιπλέον θα γίνει μέριμνα για τον συγχρονισμό των πινάκων της βάσης δεδομένων νόσησης με αυτή των ατόμων που βρέθηκαν σε ελεγχόμενες περιοχές και συγγραφή σχετικού αλγορίθμου. Ο αλγόριθμος θα αποτελεί κώδικα που θα ελέγχει και θα υπολογίζει μέσω συνθηκών τον τόπο και χρόνο παραμονής σε ελεγχόμενη περιοχή ατόμου που θετικοποιήθηκε συγκρίνοντας και υπολογίζοντας κατά πόσο συμπίπτουν με τον αντίστοιχο τόπο και χρόνο οποιοδήποτε άλλων χρηστών. Σε περίπτωση αληθών συνθηκών θα αποστέλλεται το σχετικό SMS στους αντίστοιχους χρήστες τηλεφωνικών συνδέσεων.

Η δεύτερη ομάδα θα απασχοληθεί με το front-end της εφαρμογής, δηλαδή την ανάπτυξη του λογισμικού που αφορά το γραφικό περιβάλλον (interface) του χρήστη. Επιπλέον θα φροντίσει για τη σωστή αποκωδικοποίηση των πληροφοριών σε QRcode και τη συσχέτιση με την ανάλογη κατάσταση νόσησης. Δεδομένου ότι η εφαρμογή θα χρησιμοποιείται σχεδόν από όλους τους πολίτες, η διεπαφή της εφαρμογής με τους χρήστες θα πρέπει να είναι απλή και φιλική. Επίσης θα μεριμνήσει για την ορθή εγκατάσταση της εφαρμογής από τους χρήστες στις πλατφόρμες android & ios και θα μεριμνεί για την προσεκτική παρακολούθηση και ανάλυση των σχολίων των χρηστών που αφορούν την λειτουργία της εφαρμογής. Επιπροσθέτως θα πραγματοποιούνται από την ομάδα δοκιμές χρήσης της εφαρμογής για εντοπισμό σφαλμάτων κατά την χρήση της.

Η τρίτη ομάδα θα αναλάβει τον ασφαλή διαμοιρασμό των δεδομένων και την εφαρμογή όλων των μέτρων και πρωτοκόλλων που πρέπει να εφαρμοστούν στα δίκτυα και στους servers που θα χρησιμοποιηθούν, προκειμένου να αποφευχθεί παραβίαση των προσωπικών δεδομένων από τυχόν επιθέσεις κακόβουλων χρηστών (cyber attacks). Θα υπάρξει μέριμνα για την κρυπτογράφηση των πληροφοριών καθώς και την εγκατάσταση των απαραίτητων λογισμικών για ανίχνευση τυχόν επιθέσεων και ευπαθειών, το οποίο θα γίνεται με τις κατάλληλες δοκιμές και τέστ (penetration testing). Θα πραγματοποιείται σε τακτική βάση ανάλυση των τρωτών σημείων και μελέτη για τυχόν ανανέωση ή και εγκατάσταση επιπρόσθετου software ή hardware (radius servers, crypto racks etc). Το ζήτημα αυτό είναι πολύ σημαντικό καθώς τυχόν διαρροή δεδομένων και πληροφοριών θα εκθέσει σημαντικά την ακεραιότητα και εμπιστοσύνη των

πολιτών, την αξιοπιστία της ομάδας λογισμικού και κατ' επέκταση το κύρος της εκάστοτε κυβέρνησης ζημιώνοντας με χρηματικά πρόστιμα και ποινικές ευθύνες. Επίσης θα προσαρμόσει την ανάπτυξη του λογισμικού στα απαραίτητα πρωτόκολλα για την πιστή τήρηση και εφαρμογή του General Data Protection Regulation (GDPR) προς αποφυγή κυρώσεων και επιβολής προστίμων. Θα μελετούν συνεχώς την εξέλιξη και τυχόν ανανέωση του GDPR και θα επιβλέπουν την τήρηση και την εφαρμογή του σε όλα τα στάδια και τα πεδία της εφαρμογής.

Η επικοινωνία μεταξύ των ομάδων δύναται να πραγματοποιείται, με online συνδιαλέξεις, εφαρμογές άμεσων μηνυμάτων, μέσα κοινωνικής δικτύωσης και με παροχές απομακρυσμένης πρόσβασης.

Scrum Master: Είναι υπεύθυνος για τη διασφάλιση της ορθής ανάπτυξης του έργου και βοηθά τα μέλη της ομάδας να μάθουν και να εφαρμόσουν καλύτερα τις πρακτικές που ακολουθούνται στην διαδικασία του Scrum. Κάνει ό,τι μπορεί ώστε να βοηθήσει την ομάδα και τον Product Owner. Επειδή τα εμπόδια και οι δυσκολίες είναι αναπόφευκτα, είναι σημαντικό να υπάρχει κάποιος που να μπορεί να δώσει ουσιαστικές λύσεις, έτσι ώστε η ομάδα να διατηρήσει όσο το δυνατόν ψηλότερα το επίπεδο παραγωγικότητας. Θα πρέπει να ενημερώνεται για τις αλλαγές που μπορεί να προκληθούν από το εξωτερικό περιβάλλον, για τα υγειονομικά πρωτόκολλα, τις απαιτήσεις των χρηστών, τον εξοπλισμό που θα απαιτηθεί, τον συγχρονισμό και την υλοποίηση των καταστάσεων σχετικά με τον χρονοπρογραμματισμό του έργου.

Product Owner: Είναι ο υπεύθυνος για την οικονομική αποδοτικότητα της ανάπτυξης του προϊόντος. Εντοπίζει τα χαρακτηριστικά του προϊόντος, και τα καταγράφει σε μια λίστα απαιτήσεων κατά προτεραιότητα. Ο Product Owner έχει καθοριστικό ρόλο στην ανάπτυξη του λογισμικού. Στην περίπτωση του covidpass μπορεί να είναι και αρμόδιος του εκάστοτε υπουργείου που έχει οριστεί ως υπεύθυνος του έργου ανάπτυξης του λογισμικού. Πρέπει να σημειωθεί ότι στη Scrum τον ρόλο του Product Owner μπορούν να τον αναλάβουν και περισσότερα από ένα άτομο.

3.2.2 Διαδικασία

Ο Product Owner ορίζει μια λίστα με τα χαρακτηριστικά που θέλει να ενσωματωθούν στο προϊόν η οποία ονομάζεται Product Backlog (λίστα εργασιών που αφορούν το προϊόν και δεν έχουν εκτελεστεί). Η λίστα ενημερώνεται κατά τη διάρκεια ανάπτυξης του λογισμικού και περιέχει τις απαιτήσεις ανάπτυξής του. Οι απαιτήσεις προέρχονται είτε από τους χρήστες είτε από τους υπεύθυνους για την ανάπτυξη του λογισμικού. Οι απαιτήσεις κατατάσσονται ανάλογα με την προτεραιότητά τους και υπολογίζεται ο χρόνος και η εργασία που απαιτούνται για την υλοποίησή τους. Η λίστα με τις απαιτήσεις ενημερώνεται και αναθεωρείται σε κάθε επανάληψη με καινούρια και πιο λεπτομερή στοιχεία, καθώς επίσης και με ακριβέστερες εκτιμήσεις για την αναγκαία προσπάθεια και τις νέες προτεραιότητες που θα προκύψουν.

Το Product Backlog περιλαμβάνει τα χαρακτηριστικά του λογισμικού – προϊόντος, όπως αυτά ορίζονται από τον οργανισμό και τις απαιτήσεις των πολιτών, αλλά μπορεί να περιλαμβάνει και βελτιώσεις ή σφάλματα που θα πρέπει να τύχουν προσοχής. Τα χαρακτηριστικά που περιλαμβάνονται στη λίστα, μπορούν να διατυπωθούν με οποιονδήποτε τρόπο, όπως για παράδειγμα μέσω σεναρίων Use cases ή User stories. Το σύνολο των χαρακτηριστικών του Product Backlog που επιλέγονται για μια επανάληψη (sprint), ονομάζεται Release Backlog. Η σειρά με την οποία επιλέγονται προς υλοποίηση εξαρτάται από την προτεραιότητα που έχει δοθεί σε αυτά. Συνήθως επιλέγονται πρώτα χαρακτηριστικά με μεγάλη αξία και λιγότερη απαιτούμενη προσπάθεια για υλοποίηση. Σχετικά με την εκτιμώμενη προσπάθεια που απαιτείται για την υλοποίηση του κάθε χαρακτηριστικού της λίστας, η ομάδα Scrum ενημερώνει τον Product Owner.

Στον πίνακα 1 που ακολουθεί παρουσιάζεται το product backlog του CovidPass (επισυνάπτεται και σε αρχείο excel)

PRODUCT_BACKLOG_COVIDPASS.xlsx					
	A	B	C	D	E
1					
2	PRODUCT BACKLOG COVIDPASS				
3					
4					
5	USER STORY ID	USER STORY	PRIORITY	SPINT	TASK OWNER
6	US001	Ως χρήστης θέλω να κατέβασω την εφαρμογή	5	1	ΡΑΦΑΗΛ ΚΑΤΡΑΜΑΔΟΣ
7	US002	Ως χρήστης θέλω να κάνω login στην εφαρμογή	1	1	ΑΓΓΕΛΙΚΗ ΑΓΓΕΛΙΔΟΥ
8	US003	Ως χρήστης θέλω να κάνω checkin σε δημοσιο χώρο	1	1	ΠΑΥΛΟΣ ΠΑΥΛΙΔΗΣ
9	US004	Ως χρήστης θέλω να κάνω checkout απο δημοσιο χώρο	2	2	ΘΑΝΑΣΗΣ ΑΘΑΝΑΣΙΟΥ
10	US006	Ως χρήστης θέλω να δω τις οδηγίες χρήσης	5	3	ΑΡΓΥΡΗΣ ΑΡΓΥΡΙΟΥ
11	US007	Ως ελεγκτής θέλω να σκαναρω το QR CODE	4	3	ΘΑΝΑΣΗΣ ΑΘΑΝΑΣΙΟΥ

(πίνακας 1: Product backlog)

Ως user story, όπως αναφέρθηκε και παραπάνω, αποτελεί η περίπτωση χρήσης που πρόκειται να μελετηθεί. Στη συνέχεια αναφέρεται ο βαθμός προτεραιότητας υλοποίησης του ζητήματος, ο κύκλος sprint που πρόκειται να αναπτυχθεί – μελετηθεί και ο task owner δηλαδή ο developer που πρόκειται να αναλάβει την διεργασία.

3.2.3 Προγραμματισμός ενός Sprint (Sprint Planning)

Στην αρχή του κάθε Sprint, πραγματοποιείται το Sprint Planning Meeting. Η συνεδρίαση χωρίζεται σε δύο μέρη τα οποία απαντούν στις παρακάτω ερωτήσεις:

- Τι θα παραδοθεί στον οργανισμό στο τέλος του Sprint;
- Πώς θα προγραμματιστούν οι εργασίες ώστε να επιτευχθεί η παράδοση;

Το πρώτο μέρος του Sprint Planning Meeting επικεντρώνεται στην κατανόηση του τι ακριβώς ζητά ο Product Owner και κατά τη διάρκειά του διαπραγματεύεται το τι θα υλοποιηθεί και επανεξετάζονται τα χαρακτηριστικά με την μέγιστη προτεραιότητα και με μεγάλη αξία που βρίσκονται στο Product Backlog. Οι αρμόδιοι αυτής της διαδικασίας θέτουν ζητήματα σχετικά με την υλοποίηση των στόχων και τις απαιτήσεις που θα υλοποιηθούν.

Το δεύτερο μέρος της σύσκεψης επικεντρώνεται στον σχολαστικό προγραμματισμό των διεργασιών (Tasks) που πρέπει να γίνουν. Τα μέλη της ομάδας επιλέγουν τα χαρακτηριστικά από την λίστα εργασιών (Product Backlog) και δεσμεύονται να τα ολοκληρώσουν έως το τέλος του Sprint, ξεκινώντας από την αρχή της λίστας, δηλαδή από εκείνα που χαρακτηρίστηκαν με την μεγαλύτερη προτεραιότητα.

Η μορφή των user stories πρέπει να είναι της μορφής: Ως “τύπος χρήστη”, θέλω να κάνω “μια ενέργεια”, ώστε να επιτευχθεί μια διεργασία.

ΕΡΓΑΣΙΑ ΚΑΤΡΑΜΑΔΟΥ ΡΑΦΑΗΛ – ΣΥΣΤΗΜΑ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ COVID19

Δεδομένης της επιδημιολογικής κρίσης και της γρήγορης ανάπτυξης του λογισμικού που απαιτείται για την πρόληψη της διασποράς του ιού covid19 καθώς και την αποφυγή lockdown όπως επίσης και της λειτουργίας των δημοσίων χώρων, η ανάπτυξη του covidpass θα γίνει σε τρία sprint των δυο εβδομάδων (14 ημερών). Στους παρακάτω πίνακες (πίνακας 2,3 και 4) παρουσιάζεται η εφαρμογή των sprints backlogs. (επισυνάπτεται και σε αρχείο excel)

SPRINT1.xlsx

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	SPRINT [1] BACKLOG																			
2																				
3	ID	User Story	Tasks	Owner	Status	Estimate d effort	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
4	US001	Ως χρήστης θέλω να κατέβασω την εφαρμογή	MODIFY DATABASE	ΡΑΦΑΗΛ ΚΑΤΡΑΜΑΔΟΣ	Completed	16	0.5	0	0	0	0	0	0	0	0	0	0	0	0	0
5			CREATE UI	ΑΓΓΕΛΙΚΗ ΑΓΓΕΛΙΔΟΥ	Completed		0	8	0	0	0	0	0	0	0	0	0	0	0	0
6			INSTALL LOGIN WIDGET ON PLATFORM	ΡΑΦΑΗΛ ΚΑΤΡΑΜΑΔΟΣ	Completed		0	0	6	2	0	0	0	0	0	0	0	0	0	0
7	US002	Ως χρήστης θέλω να κανω login στην εφαρμογή	CREATE JAVASCRIPT LOGIN	ΑΓΓΕΛΙΚΗ ΑΓΓΕΛΙΔΟΥ	Completed	48	0	0	5	7	2	8	6	6	5	0	0	0	0	0
8			MODIFY DATABASE WITH GRADIENT	ΡΑΦΑΗΛ ΚΑΤΡΑΜΑΔΟΣ	Completed		0	0	0	0	0	0	0	0	2	3	0	0	0	0
9																				
10	total					64														

(πίνακας 2)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	SPRINT [2] BACKLOG																			
2																				
3	ID	User Story	Tasks	Owner	Status	Estimate d effort	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
4	US003	Ως χρήστης θέλω να κανω checkin σε δημοσιο χωρο	MODIFY DATABASE & CHECK SECURITY RULES	ΠΑΥΛΟΣ ΠΑΥΛΙΔΗΣ	Completed	15	2	0	0	0	0	3	0	0	0	0	0	0	0	0
5			CREATE UI	ΘΑΝΑΣΙΗΣ ΑΘΑΝΑΣΙΟΥ	Completed		0	2	0	0	0	0	0	0	0	0	0	0	0	0
6			INSTALL CHECK IN WIDGET ON PLATFORM	ΠΑΥΛΟΣ ΠΑΥΛΙΔΗΣ	Completed		0	0	6	2	0	0	0	0	0	0	0	0	0	0
7	US004	Ως χρήστης θέλω να κανω checkout απο δημοσιο χωρο	CREATE JAVASCRIPT LOGOUT	ΘΑΝΑΣΙΗΣ ΑΘΑΝΑΣΙΟΥ	Completed	44	0	0	5	7	2	8	6	6	5	0	0	0	0	0
8			MODIFY DATABASE	ΠΑΥΛΟΣ ΠΑΥΛΙΔΗΣ	Completed		0	0	0	0	0	0	0	0	2	3	0	0	0	0
9					total	59														
10																				

(πίνακας 3)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	SPRINT [3] BACKLOG																			
2																				
3	ID	User Story	Tasks	Owner	Status	Estimate d effort	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14
4	US004	Ως χρήστης θέλω να δω τις οδηγίες χρήσης	CHECK SECURITY RULES	ΑΡΓΥΡΗΣ ΑΡΓΥΡΙΟΥ	Completed	22	1	0	0	2	0	3	0	0	0	0	0	0	0	0
5			CREATE UI	ΚΩΣΤΑΣ ΚΩΝΣΤΑΝΤΙΟΥ	Completed		0	2	0	0	0	2	0	0	0	0	0	5	0	0
6			WRITE DIRECTION & MANUAL GUIDE	ΑΡΓΥΡΗΣ ΑΡΓΥΡΙΟΥ	Completed		0	0	2	2	0	0	0	0	1	0	0	2	0	0
7	US006	Ως ελεγκτής θέλω να σκαναρω το QR CODE	CREATE ALGORITHM CHECK	ΚΩΣΤΑΣ ΚΩΝΣΤΑΝΤΙΟΥ	Completed	44	0	0	3	7	2	4	6	6	2	0	0	0	0	0
8			MODIFY CAMERA SWITCH	ΑΡΓΥΡΗΣ ΑΡΓΥΡΙΟΥ	Completed		0	0	1	0	0	0	0	0	2	3	0	0	0	0
9					total	66														
10																				

(πίνακας 4)

Στα Sprint Backlog^{1,2,3} που παρουσιάστηκαν παραπάνω, στην πρώτη στήλη φαίνονται τα χαρακτηριστικά τα οποία επιλέχθηκαν από το Product Backlog (ID) και διασπάστηκαν σε επιμέρους εργασίες, οι οποίες φαίνονται στη τρίτη στήλη (Tasks). Στη δεύτερη στήλη παρουσιάζεται το user story, δηλαδή η υποθετική ενέργεια του χρήστη. Στην τέταρτη στήλη εμφανίζονται τα μέλη της ομάδας στα οποία έχουν ανατεθεί οι εργασίες. Στην πέμπτη στήλη παρουσιάζεται η εκτίμηση για την προσπάθεια που απαιτείται για την ολοκλήρωση κάθε εργασίας σε ώρες. Για την εφαρμογή covidpass, η συνολική εκτίμηση για την προσπάθεια που απαιτείται για το 1^ο sprint είναι 64 ώρες, 2^ο sprint είναι 59 ώρες και 3^ο sprint είναι 66 ώρες.

3.2.4 Καθημερινή συνεδρίαση Scrum (Daily Scrum)

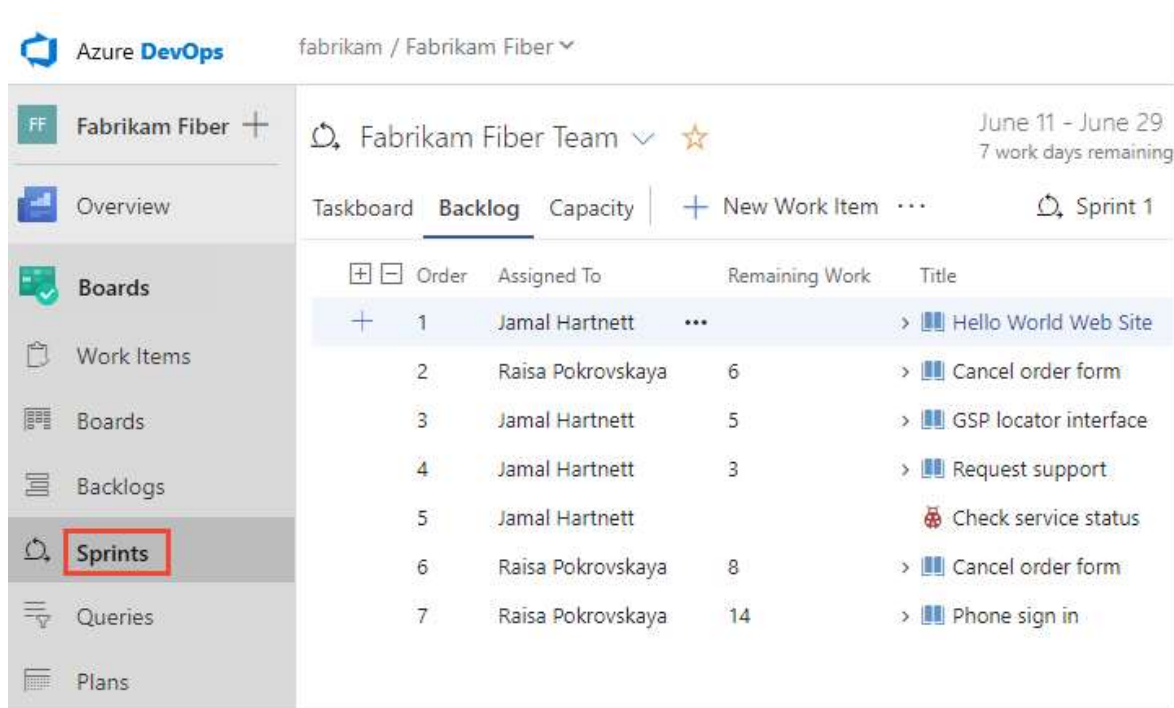
Η ομάδα συμμετέχει σε καθημερινές συναντήσεις – meeting που πραγματοποιούνται κατά την διάρκεια ενός Sprint και ονομάζονται «The Daily Scrum Meetings». Η διάρκεια του κάθε meeting δεν μπορεί να υπερβαίνει τα δεκαπέντε λεπτά. Στη σύσκεψη αυτή συμμετέχει όλη η ομάδα ώστε τα μέλη της να συγχρονιστούν και συντονιστούν μεταξύ τους, με θέμα συζήτησης τυχόν προβλήματα και δυσκολίες που υπάρχουν. Κατά τη σύσκεψη αυτή κάθε μέλος της ομάδας πρέπει να κοινοποιήσει τρία βασικά πράγματα:

- Καινούριες ενέργειες που έγιναν μετά την τελευταία σύσκεψη.
- Εμπόδια και δυσκολίες και ο τρόπος που αντιμετωπίστηκαν.
- Στόχος μέχρι το επόμενο Daily Scrum Meeting

Στο Daily Scrum meeting θα πρέπει να παρευρίσκονται μόνο η ομάδα Scrum και ο Scrum Master. Ο τόπος και ο χρόνος της σύσκεψης θα πρέπει να είναι προκαθορισμένος και ο ίδιος κάθε φορά. Σκοπός της είναι να βελτιωθούν η συνεργασία, η λήψη αποφάσεων και η επίλυση προβλημάτων.

3.2.7 ΔΙΑΧΕΙΡΙΣΗ SCRUM ME CLOUD COMPUTING

Η εξέλιξη του cloud computing μπορεί να μας δώσει τη δυνατότητα της διαχείριση έργων με scrum σε όλο τον κύκλο ζωής ανάπτυξης του λογισμικού. Εργαλεία που είναι διαθέσιμα σε πλατφόρμες cloud computing όπως το Microsoft Azure boards επιτρέπει στην ομάδα να παρακολουθεί τα tasks, την κατάσταση εργασίας, user stories, τις εκκρεμότητες, τις δυνατότητες και να παρακολουθεί τα bugs και ελαττώματα που θα παρατηρηθούν και θα προκύψουν κατά την ανάπτυξη και διαχείριση του έργου – λογισμικού.



Εικόνα 14 Sprint backlogs and taskboards overview

πηγή: <https://docs.microsoft.com/en-us/azure/devops/boards/sprints/scrum-overview?view=azure-devops>)

4. Περιγραφή των καταστάσεων λειτουργίας – Περιπτώσεις χρήσης

Στο κεφάλαιο που ακολουθεί παρουσιάζονται οι περιπτώσεις χρήσης (use case) της εφαρμογής για τις τρεις καταστάσεις. Πράσινη ένδειξη: Χωρίς μόλυνση Covid19. Πορτοκαλί ένδειξη: υποψία μόλυνσης Covid19. Κόκκινη ένδειξη: Επιβεβαιωμένη μόλυνση Covid19.

Το «X» άτομο ελέγχεται μία ημέρα μετά την έξοδο του από κλειστό δημόσιο χώρο διαγιγνώσκεται θετικό στον covid19:

pre-conditions (για όλες τις περιπτώσεις):

1. Ο χρήστης είναι κάτοχος ΑΦΜ και έχει κάνει εγγραφή σε προγενέστερο χρόνο στην ιστοσελίδα του φορολογικού μητρώου.
 2. Έχει καταχωρήσει στην προαναφερόμενη ιστοσελίδα τον αριθμό της τηλεφωνικής του σύνδεσης για την οποία είναι χρήστης.
 3. Είναι κάτοχος smartphone ή tablet.
 4. Έχει κατεβάσει την εφαρμογή από τις προεγκατεστημένες πλατφόρμες Playstore ή IOS
 5. Έχει σύνδεση και πρόσβαση στο διαδίκτυο.
- Ο χρήστης έχει κάνει σε προγενέστερο χρόνο LOGIN στην εφαρμογή.
 - Πραγματοποίηση CHECKIN και επιλογή δημόσιου χώρου που πρόκειται να εισέλθει.
 - Καταγραφή – καταχώρηση τόπου και χρόνου του ατόμου που έχει εισέλθει.
 - Μία ημέρα μετά διαγιγνώσκεται θετικός στον Covid19.
 - Ενημέρωση της κατάστασης όλων των ατόμων που βρίσκονταν στον ίδιο τόπο και χρόνο με SMS..
 - Απομόνωση για 14 ημέρες και αλλαγή κατάστασης από πράσινη σε κόκκινη.

- Αλλαγή κατάστασης από πράσινη σε πορτοκαλί σε όσους είχαν έρθει σε επαφή μαζί του στον ίδιο κλειστό δημόσιο χώρο κατά τα οριζόμενα για την εφαρμογή covidpass (εντός 2 ημερών και είχαν συνάντηση για περισσότερο από 15 λεπτά).
- Πράσινη ένδειξη μετά το πέρας των 14 ημερών.

Post-conditions:

1. Ο χρήστης έχει κάνει σωστά checkout. Δηλαδή την ακριβή ώρα που εξήλθε από τον κλειστό δημόσιο χώρο

Το «X» άτομο έχει έρθει σε επαφή με θετικό κρούσμα στον ίδιο κλειστό χώρο εντός 2 ημερών και είχαν συνάντηση για περισσότερο από 15 λεπτά.

- Ο χρήστης έχει κάνει σε προγενέστερο χρόνο LOGIN στην εφαρμογή.
- Πραγματοποίηση CHECKIN και επιλογή δημόσιου χώρου που πρόκειται να εισέλθει.
- Έρχεται σε επαφή με άτομο στον ίδιο κλειστό χώρο που μια ημέρα μετέπειτα διαγιγνώσκεται θετικό στον covid19.
- Ενημέρωση με σχετικό μήνυμα SMS στον αριθμό της τηλεφωνικής σύνδεσης για την οποία ταυτοποιήθηκε ως χρήστης.
- Πορτοκαλί ένδειξη κατά τον έλεγχο και συνιστώμενη αυτο-απομόνωση για 14 ημέρες.
- Αναίρεση από πορτοκαλί σε πράσινη ένδειξη σε περίπτωση αρνητικού covid test.
- Κόκκινη ένδειξη σε περίπτωση θετικού covid test.
- Πράσινη ένδειξη μετά το πέρας των 14 ημερών του θετικού τέστ ή της επαφής με το θετικό κρούσμα.

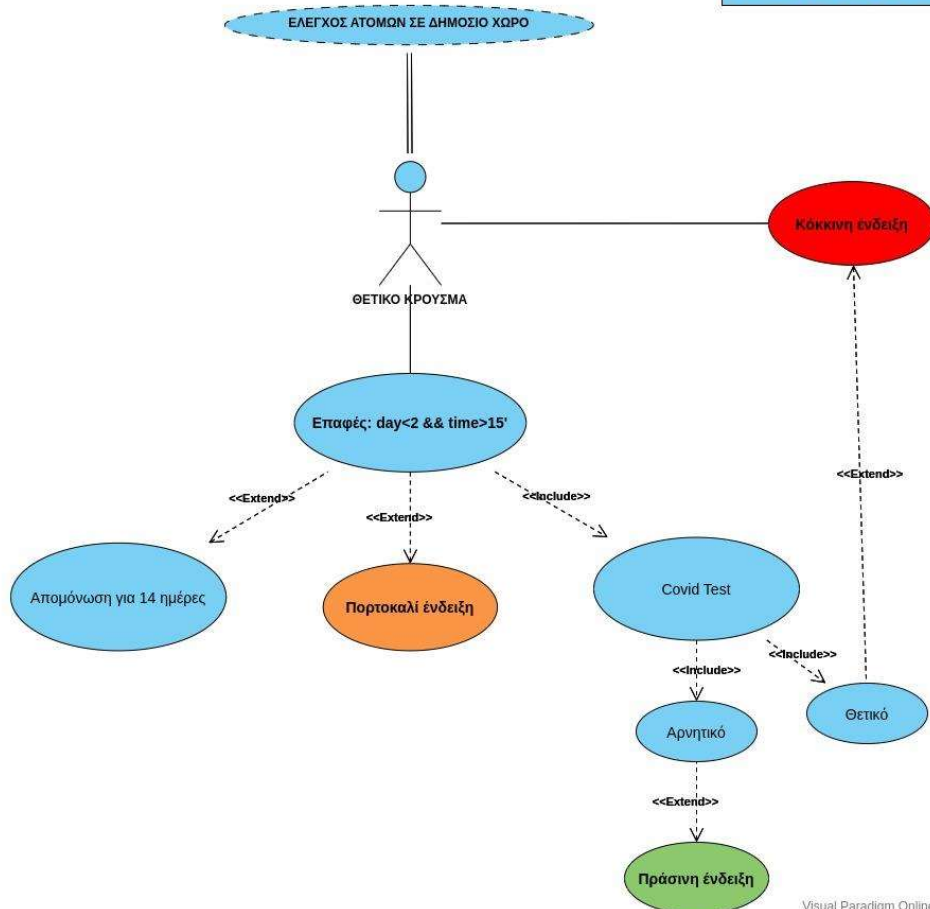
Post-conditions και pre-conditions καταστάσεις οι ίδιες όπως ορίστηκαν στην πρώτη περίπτωση

Το «X» άτομο έχει εισέλθει σε κλειστό δημόσιο χώρο με πράσινη ένδειξη.

- Ο χρήστης έχει κάνει σε προγενέστερο χρόνο LOGIN στην εφαρμογή.
- Πραγματοποίηση CHECKIN και επιλογή δημόσιου χώρου που πρόκειται να εισέλθει.
- Πραγματοποίηση CHECKOUT κατά την έξοδο του από το δημόσιο χώρο, καταγραφή στη βάση δεδομένων, τόπου και χρονικής διάρκειας παραμονής.

Post-conditions και pre-conditions καταστάσεις οι ίδιες όπως ορίστηκαν στην πρώτη περίπτωση

Στο διάγραμμα 3 που ακολουθεί παρουσιάζεται use case diagram για τις τρεις καταστάσεις λειτουργίας της εφαρμογής.



Visual Paradigm Online Free Edition

(Διάγραμμα 3: use case diagram)

5. Συμπεράσματα - εκτιμήσεις

Κατά την μελέτη ανάπτυξης του λογισμικού covidpass δίνεται έμφαση στον κύκλο ζωής λογισμικού και στον τρόπο οργάνωσης της ομάδας που πρόκειται να απασχοληθεί με το έργο. Η επικοινωνία και η οργάνωση που συνοδεύει την εφαρμογή της μεθοδολογίας scrum έχει θετικά αποτελέσματα στα μέλη και στις ομάδες που έχει κατανεμηθεί. Τα μέλη των ομάδων δεν χάνουν χρόνο και παίρνουν άμεσες απαντήσεις για τις απαιτήσεις που χρειάζονται. Υπάρχει πρόσβαση στα κατάλληλα άτομα τη σωστή χρονική στιγμή χωρίς διακρίσεις. Τα μέλη των ομάδων εμπλέκονται σε διάφορες τεχνικές και εφαρμόζουν διαφορετικές δεξιότητες χωρίς να περιορίζονται σε συγκεκριμένο ρόλο.

Η οργάνωση και ο συντονισμός της ανάπτυξης της εφαρμογής σε συνδυασμό με το φιλικό προς τον χρήστη περιβάλλον εκτιμάται πως θα είναι αποτελεσματικά ως προς την χρήση της εφαρμογής άλλα και την διαχείριση της πανδημίας, τον περιορισμό των lockdown και την “ελάφρυνση” του εθνικού συστήματος υγείας καθώς και το αίσθημα ασφάλειας των πολιτών.

6. Βιβλιογραφία

Websites –Books

1. Τεχνολογία Λογισμικού, Γιακουμάκης Μανώλης, Μάιος 2018.
2. Ο Οδηγός του Scrum, Ken Schwaber, Jeff Sutherland 2017.
3. Διαχείριση πληροφοριακών συστημάτων, Κλειδάριθμος 9/2008.

4. Scacchi, W., Models of Software Evolution: Life Cycle and Process, SEI Curriculum Module SEI-CM-10-1.0, Carnegie Mellon University, Software Engineering Institute, 1987.
5. Boehm, B. W., A Spiral Model of Software Development and Enhancement, IEEE Computer, May 1988, pp. 61-72, 1988.
6. Ειδικός ανάπτυξης λογισμικού η προγραμματιστή – Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών – Πρόγραμμα συμπληρωματικής εκπαίδευσης
https://www.researchgate.net/publication/355474411_Eisagoge_sten_technologia_logismikou_e-book.
7. Προγραμματιστικά περιβάλλοντα – Δημιουργία εφαρμογών, ebook
<http://ebooks.edu.gr/ebooks>.
8. UML διαγράμματα κλάσεων Τμήμα Πληροφορικής και Τηλεπικοινωνιών (Άρτα) Πανεπιστήμιο Ιωαννίνων Γκόγκος Χρήστος 2015 – ebook.
<https://eclass.upatras.gr>
9. 5η Διάλεξη - Διαγράμματα Κλάσης (1ο Μέρος).pdf – upatras, ebook.
10. https://repository.kallipos.gr/bitstream/11419/3162/2/02_chapter_02.pdf
11. <https://www.digite.com/agile/scrum-methodology/>
12. https://www.tutorialspoint.com/sdlc/sdlc_overview.htm
13. https://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm
14. <http://sprint.teilar.gr/educational/%CE%BC%CE%AD%CE%B8%CE%BF%CE%B4%CE%BF%CF%82-scrum/>
15. <https://people.iee.ihu.gr/~sfetsos/Scrum.html>
16. <https://www.techtarget.com/searchsoftwarequality/definition/systems-development-life-cycle>
17. <https://www.clouddefense.ai/blog/system-development-life-cycle>
18. <https://www.innovativearchitects.com/KnowledgeCenter/basic-IT-systems/system-development-life-cycle.aspx>
19. AZ-900x Module Core Azure Services. Μητροπολιτικό Κολλέγιο 07/05/2022, Δρ Άρης Παπακώστας, Microsoft Azure AZ-900 Fundamentals – Διάλεξη microsoft.

Software – Tools

1. <https://proto.io/>
2. Visual Paradigm - <https://www.visual-paradigm.com>
3. <https://me-qr.com/>
4. <https://creately.com/>