

Final Project

Felisa Gopez-Marston/Sadegh Bamohabbat Chafjiri

Introduction:

The purpose of this project is to develop a mobile application to assist psychologists and clinicians in documenting therapy and treatment sessions, streamline patient care, manage billing for behavioral health and utilize analytics for more effective treatments. The goal of this project is to offer mental health providers a tool that will assist in conducting their business efficiently. The business plan falls under the umbrella of Electronic Health Record (EHR).

The mission statement is to build a mobile application that will record the sessions between clinicians and patients, create transcription of the sessions and update patient records by June 2019.

Project Management Method:

This specific project will be developed in phases to reduce cycle time and deliver business benefits earlier and more frequently. The main benefits of having an agile methodology are:

1. Time-to-Market – More functionality to release quickly and obtain customer base and make money
2. Certainty – know what the end product will look like
3. Increase Revenue by focusing on customer value and implementing customer feedback to improve product

This management approach will reduce cost and impact of change. The plan is to deliver a prototype by mid-March to gather feedback from end-users and build a working application by June, 2019. The project will be divided into Phases. There will be at least 2 phases. The first phase is to enable the recording feature and automated transcription service which includes security and authentication. This will facilitate the collection of the data. The second phase is to analyze data collected and generate a dashboard that will gather insights and will be used to diagnose and treat patients. The target users for this application are mental counselors, psychologists and clinicians. Initial rollout will be limited to 20 clinicians at UCSF who will use the application.

REFER to Project Charter on Appendix

SDLC:

While the development phase will be utilizing the Agile Project Management Methodology, the requirements phase will be using the waterfall project management methodology to fully understand and document initial requirements. Some of the approaches include interviewing clinicians, observing psychologists and obtaining sample documents to create a workflow that represents their day-to-day tasks. As mentioned earlier, the design and development phase will be using the agile methodology to develop the desired functionalities in a series of manageable iterations. The benefit of this approach is to see incremental value in each iteration and to refine the solution over the iterations. The Agile Methodology will be used to develop the solution in a series of Scrum Sprints. Each sprint will be approximately 4 weeks and

addresses a specific portion of the desired product functionality. The desired product functionality is tracked as Functionality Backlog, which is progressively reduced over the iterations. A burn-down chart of backlog over time is another artifact to track the progress of solution development.

Project Team:

The project team will comprise of the following roles: The product owner will be responsible for representing everyone with a stake in the project and the its resulting system. The Scrum Master is responsible for the SCRUM process, rules and daily stand-up meetings. The project team will consist of a business/systems analyst, data modeler, solutions architect, technical architect, API developer, iPhone developer, android developer, infrastructure person and testers. This is not a traditional software engineering team. Instead, everyone on the project works together to complete a set of work they have collectively committed to complete within the sprint.

The 'Day to Day' Team will consist of:

- Scrum Master
- Product Owner
- The Project Team

The Leadership Team will consist of:

- Steering Committee
- Project Sponsor/Champion
- Delivery Manager (s)

Along with the direct team members, the external team members will be delivering product backlog which will turn into an increment of functionality. The project team commits to an iteration of work (sprint), participates in the Daily Scrum to synchronize with the team and develop a deep form of camaraderie and a feeling that “we’re all in this together.”

When planning A SCRUM project, the following is required.

- Create a vision for the project
- Project Proposal
- Create a Product backlog (with estimates)

The Product Backlog is the master list of desired product functionality that will be implemented during the development Sprints. An initial Product Backlog will be created during Sprint 0 and it will be refined as the project progresses. It is the responsibility of the Product Owner to prioritize the product backlog.

OVERVIEW of SPRINT

A sprint is a one cycle of development to create desired output functionality and is typically 4 to 5 weeks.

- Sprint Planning Session
- Takes place at the start of each Sprint
- The goal is to select the product functionality that will be developed during the Sprint
- The attendees are the Scrum Master, Product Owner, and the Project Team

Daily Scrum Meeting

- Takes place daily during the Sprint (15 min. report out)
- Each project team member provides a status on their Sprint tasks
- The attendees are the Scrum Master and Project Team

Sprint Review Session

- Takes place at the conclusion of a Sprint
- The project team demonstrates the functionality completed during the Sprint
- All project stakeholders have the options to attend

Sprint Retrospective

- Takes place after the Review Session
- The project team discusses what worked well & what process changes are needed before the next Sprint
- The attendees are the Scrum Master & the Project Team

Software Requirements Document

FUNCTIONAL

1. The application must be able to record clinician's notations and sessions with patients.
2. The application must be able to recognize and record voices in high quality in order to be converted with minimal error.

3. The application must be able to convert audio files into text files.
4. The application must be able to recognize a set of keywords or phrases that are likely to be spoken or commonly used based on context and content filtering.
5. The application must be able to store the text files (transcription files) in the cloud.
6. The application must be able to store recording sessions in audio files most preferably in MP3 format.
7. The application must be able retrieve current and past clinician recordings anytime.
8. The application must be able to store audio files for a particular patient and staff using a unique identification.

NON-FUNCTIONAL REQUIREMENTS

1. The application must be accessible via mobile or phone devices and desktop.
2. The application must be able to restrict staff to access only records belonging to patients under their care.
3. The application must able to work using Apple or Android Operating System.
4. The application must be able to integrate with the latest mobile authentication services.
5. The application must be able to store and backup all audio files and text files.

USE CASES:

Use Case 1: Any authorized XYZ CORP user must have the ability to record a session any time.

Name: Record a Session

Primary Actor(s): Clinician and the Patient

Description: This use case describes how a clinician can record a session.

Pre-Conditions: Before this action can be performed, the clinician must have a valid XYZ CORP App and App is in working condition and clinician has been set up in XYZ CORP application as a paying XYZ CORP customer.

The flow of Events:

Base Scenario

1. XYZ CORP displays “Welcome Page with XYZ CORP Logo” and prompts the user to enter credentials/PIN.
2. The user enters credentials.
3. XYZ CORP validates the user and then displays a list of actions.
4. The user chooses “Record a Session” or microphone button/icon.
5. XYZ CORP opens the app and displays the START button.
6. User presses on the Start Button and the user speaks.
7. XYZ CORP app transforms the phone into a digital audio recording and listening mode and starts recording.
8. At the beginning of the recording, the user will state important information such as the patient ID, Date, Time and type of session.

9. XYZ CORP app will record the entire audio file.
10. The user can pause the recording during the session and restart from point of pause.
11. XYZ CORP app is able to differentiate between pause, restart, stop and cancel (cancel out of the recording mode) buttons.
12. The user stops the recording by selecting STOP and XYZ CORP app prompts the user to confirm if the session is finished.
13. When the user confirms session is completed by selecting DONE or SUBMIT, XYZ CORP app saves and labels the audio files on the recording device and in backup locations (cloud storage) real-time or every 1 hour.

- Alternative Scenario(s)

8(a) If the customer decides to abandon the transaction, a cancel button is available

- Exception Scenario(s)

2(a) XYZ CORP

The customer enters the incorrect PIN and XYZ CORP asks to enter it again (max 3 times).

Post Conditions:

After this action is performed, the audio file is saved in the device and backed up on the cloud within a specified time increment.

- Data Involved:
 - o Name of customer
 - o Name of patient and id of patient
 - o Date time of the meeting
 - o Session Notes
 - o File Name
 - o Type of Session
 - o Other keywords used in therapy counseling
- Business Requirements
 - o The PIN must be 8 digits only – no alpha or no special characters allowed
 - o Service Fee will be based on the number of audio files
- Other Requirements: (Non-Functional)
 - o XYZ CORP must be able to identify fraud
 - o The customer has a limit of recording??? an audio file? or no storage or time limit
 - o XYZ CORP must be able to back up the audio file to the cloud in hourly increments

Use Case 2: System will create a transcript of the audio file.

Punctuation (eg. "question mark" gets replaced by "?")

* Keep screen on whilst dictating

* Allow inserting dictation in middle of a paragraph

* Increase speed of dictation

* Decrease APK size

* Capitalise sentences by default

* Add setting to hide button bar (useful for small screen devices)

Name: Create a Transcript

Primary Actor(s): Clinician

Description: This use case describes how a system creates a transcript.

Pre-Conditions: Before this action can be performed, there is an audio file in the database that belongs to a specific clinician. The audio file is in good quality, can be converted and is accessible by a laptop or computer user interface.

The flow of Events:

Base Scenario

1. XYZ CORP displays "Welcome Page with XYZ CORP Logo" and prompts the user to enter credentials/PIN.
2. The user enters credentials.
3. XYZ CORP validates the user and then displays a list of file content.
4. XYZ CORP displays content of the folder for all audio recordings and prompts the user to select which audio file to transcribe.
5. User sorts the audio file by date or by the label and selects the audio file and hits transcribe.

6. XYZ CORP converts the audio file into text or specific format and creates a text file with the same label (depending on the size of the audio file.)
7. The user chooses a text file icon next to the audio file and selects the review icon.
8. XYZ CORP displays text as translated from the audio file.
9. User proofreads and edits the text file before finalizing the text file. Once user finalizes, the file is saved in the database.
10. XYZ CORP archives file for future processes (insurance claims, data collection or tracking progress of treatment.

- Alternative Scenario(s)

4(a) If the customer decides to abandon the transaction, a cancel button is available

5(a) If the customer decides to change the audio file to be transcribed, a cancel button is available.

8(a) If the customer decides to view the text file after

- Exception Scenario(s)

2(a) XYZ CORP

Post Conditions:

After this action is performed, the text file is saved in the database in certain formats or forms.

- Data Involved:

- o Name of customer
- o Name of Clinician
- o Various Medical Reports
- o Prescription
- o History of Treatment
- o ICD 9/10 bill codes

System Design and Architecture

Cloud Architecture: Flexibility and Autoscaling

- o Stateless REST architecture, service-oriented
- o Separation of responsibilities, fault-tolerant resiliency

AI and Machine Learning

- o Sentiment analysis using Google Natural Language APIs
- o Speech-To-Text on mobile application

Security

- o OWASP Best Practices
- o Robust Security similar to banks due to HIPAA

User Experience

- o A unified experience across mobile and web

A prototype of the application is a critical deliverable to showcase the interaction between the clinician and the application/device.

REFER to Workflow on Appendix

REFER to Sequence Diagram on Appendix

REFER to Data Model on Appendix

Implementation

This is a new development or mobile application that will be utilizing a combination of Open Source Development using Google API and Services offered as SaaS and IaaS as mentioned in the proposed technology stack above. Application architecture will be decided by the Technical Architect.

The mobile application will have the following features:

- o REST Communication with backend
- o Integration with maps and geolocation for the cloud
- o Text-to-Speech

Sprint 1

Secure Login and Account Access

Account Setup

Sprint 2

Voice Recording and Transcript

Sprint 3

Store Audio Files/MPS3

REFER to Architecture on Appendix

| | TASK NAME | Release | Sprint | Sprint Ready | Story Type | Story Priority | Story Status | Story | Assigned to SPRINT | Resource Name |
|---|-----------------------------|---------|--------|--------------|------------|----------------|--------------|-------|--------------------|---------------|
| 1 | | | | | | | | | | |
| | Create Login Page | 1 | 1 | Yes | Team | High | Completed | 1 | Yes | John |
| | Create Account Page | 1 | 1 | Yes | Business | High | Completed | 3 | Yes | Jill |
| | Authenticate User | 1 | 1 | Yes | Business | High | Completed | 5 | Yes | Mark |
| | Create Terms and Conditions | 1 | 1 | Yes | Business | High | Completed | 7 | Yes | Bob |
| | Accept Payment | 1 | 1 | No | Business | High | In Prog | 9 | Yes | Mary |
| 2 | | | | | | | | | | |
| | Record Voice | 1 | 2 | Yes | Business | High | Not Started | 10 | No | |
| | Upload Recordings | 1 | 2 | Yes | Business | High | | 15 | No | |
| | Convert Recoding to Text | 1 | 2 | Yes | Business | High | | 20 | No | |
| | Listen to Recordings | 1 | 2 | No | Team | Low | | 17 | No | |
| | Select Recordings | 1 | 2 | No | Team | Low | | 18 | No | |
| 3 | | | | | | | | | | |
| | Test Login Page | 1 | 3 | Yes | Business | High | | 30 | No | |
| | Test Record | 1 | 3 | Yes | Business | High | | 31 | No | |
| | Test Upload | 1 | 3 | Yes | Business | High | | 36 | No | |

| | | | | | | | | | |
|------------------------------|---|---|-----|----------|--------|--|----|----|--|
| Test Text (transcription) | 1 | 3 | Yes | Business | High | | 40 | No | |
| Create Website | 1 | 3 | Yes | Team | Medium | | 50 | No | |
| | | | | | | | | | |

Change Management Process

Since this is a new development or mobile application using AGILE, managing change is part of the project work. The type of changes that will be accepted depends upon the following;

1. Customer Feedback –At the end of a 4-week sprint, the application is demonstrated to the users to gather feedback. If the feedback affects the competitive edge of the company, this change request is assessed and implemented.
2. Product Backlog Priorities – During the development, if the product owner realizes that some of the features are higher in priority than others, this will necessitate the changes in the implementing the features.
3. Project Team Feedback – During the course of the development, if the project team discovers new features that are critical to the success of the project, these new features will be built as part of the application
4. Missed or Misinterpreted Requirements – Due to environmental factors, technological factors or review of user stories, it is possible to accept these types of changes.

All of these changes will necessitate a Change Request form which will include an impact analysis section describing the impact to the project.

REFER to Change Request Template on Appendix

Testing:

According to the www.softwaretestinghelp.com, Quality Assurance ensures that the approaches, techniques, methods, and processes are designed for the projects are implemented correctly. QA verifies the process are followed and deliverables are created. On the other hand, Quality Control focuses on finding errors and anomalies. It is the process by which QA verify that the project deliverables meet the defined quality standards. In short, QA is the process of creating the deliverables while QC is the process to verifying that deliverables. Both are needed in any software development project.

QA Strategy

Since this is a brand-new application, Quality plans and QA procedures will have to be established and enforced during the SDLC to produce a high-quality software. This will include the required processes and documents (requirements, design, and code) and ensuring processes are followed and documents generated are within standards specified and preferably using the standard templates. For each phase/sprint, it is expected that there will be minimum documentation. In this project, there will be 1-2 people in charge of testing the quality of the application but the development team will rely on one another to improve the software quality. These activities will include the design and code review inspection. As mentioned in Sommerville's text, the main focus of an Agile method of software engineering is the development of code and, the informal approach to quality management adopted in agile methods is particularly effective for

software product development where the company developing the software also controls its specification. Since the main focus is the development code, the code must be in almost perfect quality and this can only be achieved via informal communication.

Obviously, unit testing is part of the development tasks. However, writing test cases can begin as soon as the sprint functionalities have been defined. Test-driven design (TDD) is a technique that can be applied in order to produce a high-quality product in a short period. It is designed to write tests in collaboration between customer and developers to produce an application that represents the real world. For this project, TDD would be effective. For instance, for the Record Session use case, it will generate test cases such as the following tests:

- 1.) "Record Mode" will initiate the recording function and convert the device into a "listening" mode.
- 2.) Start Button will begin recording.
- 3.) Stop Button will pause recording
- 4.) Resume Button will restart recording where it left off.
- 5.) End Button will complete the recording.

| | Function | Test Name | Type | Status |
|--|--------------|---------------------------------|--------|----------------|
| | | | | (Pass or Fail) |
| | Record Voice | Test Start Button Functionality | Manual | |
| | Record Voice | Test Pause Functionality | Manual | |
| | Record Voice | Test End Functionality | Manual | |
| | Record Voice | Test Continue Functionality | Manual | |
| | Record Voice | Test Record Mode | Manual | |
| | Record Voice | Test Listen Mode | Manual | |

| | | | | |
|--|------------------|----------------------------------|-----------|--|
| | Record Voice | Test Save Mode | Manual | |
| | Record Voice | Test Delete Recording | Manual | |
| | Upload Recording | Test Manual Upload | Manual | |
| | Upload Recording | Test Manual Upload Cancel | Manual | |
| | Upload Recording | Test Manual Upload Successful | Manual | |
| | Upload Recording | Test Manual Re-Upload | Manual | |
| | Upload Recording | Test Manual Error Upload | Manual | |
| | Upload Recording | Test Automated Upload | Automated | |
| | Upload Recording | Test Automated Upload Cancel | Automated | |
| | Upload Recording | Test Automated Upload Successful | Automated | |

Release, User and Security Testing:

Test cases will be executed in QA prior to releasing to UAT to verify that features implemented in the application correctly match approved requirements prior to releasing to the customers. This includes performing requirements traceability, white box testing, and negative testing.

User acceptance testing is a standard type of testing for any project development with customer-facing application. In this project, user acceptance will occur to gain feedback from clinicians regarding the features outlined in the three sprints. Each sprint will have UAT as means of validation that the application is built based on requirements.

Hiring security experts as part of the security testing may be critical since application contains PHI and must be HIPAA compliant. In addition to having a static code analysis and regular vulnerability scanning, hiring external resources to perform penetration testing would provide the peace of mind and assurance that the contractual and compliance needs are met. Below are considerations for security testing:

1. Investigating user permissions and user access includes adding/removing users

2. Confirming the user authorization and authentications
3. Investigating the phishing scenarios rose by social engineering
4. Analyzing the resistance of cryptographic algorithms against attacks on algorithms
- 5- Analyzing the resistance of cryptographic protocols against man-in-the-middle and hijack
- 6- Analyzing the structural/architectural resistance of security and privacy solutions against combined attacks
- 7- Testing the updating process of security solution which is sending by the new version of the patch
- 9- Security risk management which is more in the business side about financial and reputational losses after a probable data breach

Version Control:

A product can change due to environmental factors and can be accommodated by extending the requirements. A good measure to be aware of changes would be to use the concept of versioning. Some changes can overlap with other parts of products and it might be impacting a product's quality. Versioning does not always mean a new functionality or new release of the product. Versioning means a change in the application code and can be defined by using the following rules according to TechTarget. There are ways to define versions in an application.

- Major - Major version is a definite release of the product. It represents major improvements, enhancements or significant changes in functionality.

- Minor - Minor version is incremented when there is an upgrade/maintenance release including minor improvements and fixes.
- Build No - Build Number is incremented when the new build is created.

A single version repository similar to ClearCase is to be used to store all versions of the application. This centralized database has built-in functionality that can do the following:

- 1- Version and release identification: implementing changes and new components or platform
- 2- Change history recording and Storage: checking functionality and other performance issues of latest version by the customer and keeping records
- 3- Define a code line to show the sequence of source code versions

Release Plan:

This process ensured that the software meets all requirement in both customer and regulation sides. In release testing, a system delivers the general expectations of speed, performance, functionalities. It is a process of validation and checking to prove that failures will not occur during the typical use of the application. It has three steps:

1. Requirements-based testing
 - Investigating the general principle
 - Validating system response with customer's requirement

2. Scenario testing

- Building a narrative story that is credible and fairly complex to test
- Running typical scenarios to develop test cases

3. Performance testing

- Testing the system for response time or speed of loading a page
- Testing the system to ensure it is stable, scalable and reliable
- Testing the system to handle the intended load or expected number of users

Future Plans (maturing and reliability)

The plan is to initially release this product to a limited number of mental health psychologists and clinicians to gather preliminary feedback and identify minor bugs. In this phase, a change request may come into play. With their feedback, the project team can improve the product before releasing to the masses. Also, there will be new ideas stemming from using the software by end-users that will create new opportunities to enhance the software. The project team can also suggest product features based on using the product.

APPENDIX

SAMPLE Project Charter

Project: Mobile Application/Personal Assistant for Therapists
Date Prepared: November 11, 2018
Version 1
Last Update

Mission Statement: To build a mobile application that will assist Therapists in documenting the therapy sessions, tracking patients progress and billing insurance or patient within a year

1. Project Charter Summary

| | | |
|--------------------------------------|--|----------------------------|
| Project Description | <i>This project is a mobile application that will automate session notes, psychological evaluations, progress notes and treatment plans to assist therapist in documenting patient records and billing patients or insurance.</i> | |
| Project Scope | <i>This project will be using Agile Project Methodology. The first phase for the projects with consists of automating the therapy sessions and creating an electronic record for a patient for documentation and billing purposes.</i> | |
| Project Milestones | Phase | Proposed Timeframes |
| | <i>Requirements</i> | <i>January 2019</i> |
| | <i>Design</i> | <i>February 2019</i> |
| | <i>Construction</i> | <i>April 2019</i> |
| | <i>QA Pass</i> | <i>May 2019</i> |
| | <i>Implementation</i> | <i>June 2019</i> |
| | <i>Support Handoff</i> | <i>July 2019</i> |
| | | |
| | | |
| Project Owner | | |
| Project Manager | | |
| Estimated Cost | \$ | |
| Project Management Approach | <i>Agile Methodology</i> | |
| Key Issues and Considerations | <i>HIPAA compliant, Strong Security, IOS and Android Platforms Supported</i> | |
| Assumptions & Constraints | <i>List key assumptions and constraints.</i> <i>Funding is established by January 2019.</i> <i>Architecture Design and Technology are finalized by February 2019.</i> <i>Resources with skills set will available to begin coding in February 2019.</i> | |

2. Project Charter Detail

2.1. Introduction and Background

The purpose of this project is to develop a mobile application to assist psychologists and clinicians in documenting therapy and treatment sessions, streamline patient care, manage billing for behavior health and utilize analytics for more effective treatments. The goal of this project is to offer mental health providers a tool that will assist in conducting their business efficiently. The business plan falls under the umbrella of Electronic Health Record (EHR).

The mission statement is to build a mobile application that will record the sessions between clinicians and patients, create transcription of the sessions and update patient records by June 2019.

Project Management Method:

This specific project will be developed in phases to reduce cycle time and deliver business benefits earlier and more frequently. The main benefits of having an agile methodology are:

1. Time-to-Market – More functionality to release quickly and obtain customer base and make money
2. Certainty – know what the end product will look like
3. Increase Revenue by focusing on customer value and implementing customer feedback to improve product

This management approach will reduce cost and impact of change. The plan is to deliver a prototype by mid-March to gather feedback from end-users and build a working application by June, 2019.

The project will be divided into Phases. There will be at least 2 phases. The first phase is to enable the recording feature and automated transcription service which includes security and authentication. This will facilitate the collection of the data. The second phase is to analyze data collected and generate a dashboard that will gather insights and will be used to diagnose and treat patients.

The target users for this application are mental counselors, psychologists and clinicians. Initial rollout will be limited to 20 clinicians at UCSF who will use the application.

2.2. Benefits and Requirements

| Benefits | Functional Requirements |
|-------------------------|---|
| <i>Business Benefit</i> | <i>How the system enables that benefit.</i> |
| | |

2.3. Scope

Agile Development will be used.

2.4. Project Management Approach

- *Waterfall for Requirements Phase*
- *Agile for Design thru Implementation Phase*
- *Fixed costs or time and materials*

2.5. Project Risks

This section includes an assessment of the most significant business, technical and execution risks that may impact the viability of the proposed investment.

| Description | Likelihood (High/Medium/Low) | Impact (High/Medium/Low) | Mitigation |
|-------------------------|--|--|--|
| <i>Describe a risk.</i> | <i>Rate the likelihood that the risk will occur.</i> | <i>Rate the impact to the project should the risk occur.</i> | <i>How will you handle it if it does happen?</i> |
| | | | |
| | | | |

2.6. Related Programs and Projects and Dependencies

| Name | Status | Dependency/Impact |
|------------------------|-------------------|---------------------------------------|
| <i>Related Project</i> | <i>In Testing</i> | <i>This project must go in first.</i> |
| | | |
| | | |

2.7. High-Level Plan and Deliverables

| Phase | Deliverables | Proposed Timeframes |
|------------------------|---|----------------------|
| <i>Requirements</i> | <i>Business Requirements Document</i> | <i>Jan 2019</i> |
| <i>Design</i> | <i>Design Document, Interface Control Specification, etc.</i> | <i>February 2019</i> |
| <i>Construction</i> | <i>Functioning Code</i> | <i>May 2019</i> |
| <i>QA Pass</i> | <i>Test Plan, Completed Test Cases, and QA Findings</i> | <i>June 2019</i> |
| <i>Implementation</i> | <i>Implementation Plan</i> | <i>July 2019</i> |
| <i>Support Handoff</i> | <i>Support Manual</i> | <i>August 2019</i> |

2.8. Deliverables and Acceptance Criteria

| Deliverables | Acceptance Criteria |
|---|---|
| <i>Business Requirements Document</i> | <i>User Signoff</i> |
| <i>Design Document, Interface Control Specification, etc.</i> | <i>Architect and Tech Lead Approval</i> |
| <i>Functioning Code</i> | <i>Tech Lead Approval</i> |
| <i>Test Plan, Completed Test Cases, and QA Findings</i> | <i>Project Manager and Tech Lead Approval</i> |
| <i>Implementation Plan</i> | <i>Team Review with No Objections</i> |
| <i>Support Manual</i> | <i>Support Team Approval</i> |

2.9. Assumptions and Constraints

- *List the assumptions and constraints. For example, We assume the existing servers will be used for this project. If we have to setup new environments, the estimates will be higher*
- *For hardware and infrastructure, this project will utilize Amazon Web Services on the Cloud*
- *For software, this project will use open source and google voice API*
- *For resources, it assumed that an experienced technical architect with mobile development experience will be available*
- *Developers with both IOS and Android skill sets will be available for the duration of the sprint(s)*

2.10. Roles and Responsibilities

| Persons | Role | Responsibilities |
|-------------|-------------------------|---|
| <i>Name</i> | <i>Project Manager</i> | <ul style="list-style-type: none">• <i>Publish the project plan</i>• <i>Conduct weekly status meetings</i>• <i>Publish meeting minutes</i>• <i>Monitor progress</i>• <i>Track and resolve issues</i> |
| <i>Name</i> | <i>Business Analyst</i> | <ul style="list-style-type: none">• <i>Write the requirements.</i>• <i>Interview subject matter experts.</i>• <i>Conduct Joint Application Development sessions.</i>• <i>Provide Use Case Tables</i>• <i>Provide Use Case Diagrams</i>• <i>Provide Activity Diagrams</i> |
| <i>Name</i> | <i>Tech Lead</i> | <ul style="list-style-type: none">• <i>Write the design document.</i>• <i>Obtain Architecture review</i>• <i>Etc.</i> |

2.11. Financial Analysis (Same Breakdown as Project Proposal)

2.11.1. Costs and Benefits

List the costs and benefits.

2.11.2. Plan for Tracking Improvements

[I'm not sure what they're looking for here]

SAMPLE CHANGE REQUEST

Change Request Document

| | | | |
|----------|----------------|---------------|--|
| Project: | 18-BI-039 | | |
| CR Name: | CR 1_18-BI-039 | | |
| CR #: | 1 | Date Created: | |
| Author: | | | |

General Information

| | | | |
|---|--|--|--|
| Change Requestor: Business Unit/Department | IT | Change Owner: Business Owner | |
| Priority: | <input type="checkbox"/> High Urgent/Show-Stopper <input type="checkbox"/> Medium Mandatory <input type="checkbox"/> Low Nice to Have | | |
| Type of Change: | <input type="checkbox"/> Add New Req. <input type="checkbox"/> Edit Existing Req. <input type="checkbox"/> Delete Existing Req. 5.1.1, 5.1.3 | | |
| Detail Description: Proposed change over current state | See Business Requirements below. | | |
| Business Value/Benefit(s): How will this change improve the current state | Provide clarification to IT and QA. | | |
| Consequence(s): Negative impact if this change is not implemented | May impact IT and QA timelines. | | |
| Technical Review Notes: Impact worthy items after Dev. review | | | |

Project Management Impact

| | | | | | |
|---------------------------|--------------------------------------|--|-------------------------------------|---|--|
| Development Effort | <input type="checkbox"/> High | <input type="checkbox"/> Medium | <input type="checkbox"/> Low | Estimated Hours: If Available | |
| QA Effort | <input type="checkbox"/> High | <input type="checkbox"/> Medium | <input type="checkbox"/> Low | Estimated Hours: If Available | |

| | | | |
|--|--|---|-------------|
| Business (BA) Effort | <input type="checkbox"/> High <input type="checkbox"/> Medium <input checked="" type="checkbox"/> Low | Estimated Hours: If Available | 2 . 0 |
| Project Timeline Impact: Brief Summary of Impact | Delay in approving CR may impact QA testing schedule. | | |

Change Control Board Decision (If Applicable) N/A

| | | |
|---|--|---|
| Date: | | Decision: <input type="checkbox"/> Approved <input type="checkbox"/> Approved with Conditions <input type="checkbox"/> Rejected <input type="checkbox"/> Postponed |
| Explanation of Decision: Required if not Approved | | |

Sponsor Decision (If Applicable)

| | | |
|---|--|---|
| Date: | | Decision: <input type="checkbox"/> Approved <input type="checkbox"/> Approved with Conditions <input type="checkbox"/> Rejected <input type="checkbox"/> Postponed |
| Explanation of Decision: Required if not Approved | | |

Business Requirements

| Requirement Number | High level Requirement | Type of Change | Detailed Requirement Number for Change |
|--------------------|---|----------------|--|
| 5.1 | The system must automatically upload the voice recording to the cloud twice a day | Edit | 5.1.1 |
| 5.2 | The system has the ability to manually upload the voice recording to the cloud. | Edit | 5.1.3 |
| | | | |
| | | | |

EDIT 5.1.1

CURRENT: The system must automatically upload the voice recording to the cloud twice a day.

PROPOSED: The system must automatically upload the voice recording to the cloud frequently depending on the account type.

| Account Type | Upload Frequency | Transcription Services | Fees |
|--------------|------------------|------------------------|------|
|--------------|------------------|------------------------|------|

| | | | |
|--------------|---|--|--|
| Premium Plus | The system must automatically upload the voice recording to the cloud every 4 hours (3 X a day) | Instant - All Files | |
| Premium | The system has the ability to manually upload the voice recording to the cloud based on configuration | Instant - Specific Files (In sequence) | |
| Standard | The system must automatically upload the voice recording to the cloud twice a day | Batch - By File | |
| Basic | The system must automatically upload the voice recording to the cloud once a day | Batch - All Files | |

REFERENCES

SOFTWARE ENGINEERING Tenth Edition by Ian Sommerville

<https://www.smartsheet.com/comprehensive-guide-values-principles-agile-manifesto>

<https://www.datamatrixmedical.com/medical-transcription-benefits/>

<https://www.liquidplanner.com/blog/5-ways-agile-helps-manage-changing-requirements/>

[google voice to text api](#)

<https://cloud.google.com/speech-to-text/>

<https://codelabs.developers.google.com/codelabs/cloud-speech-intro/index.html#0>

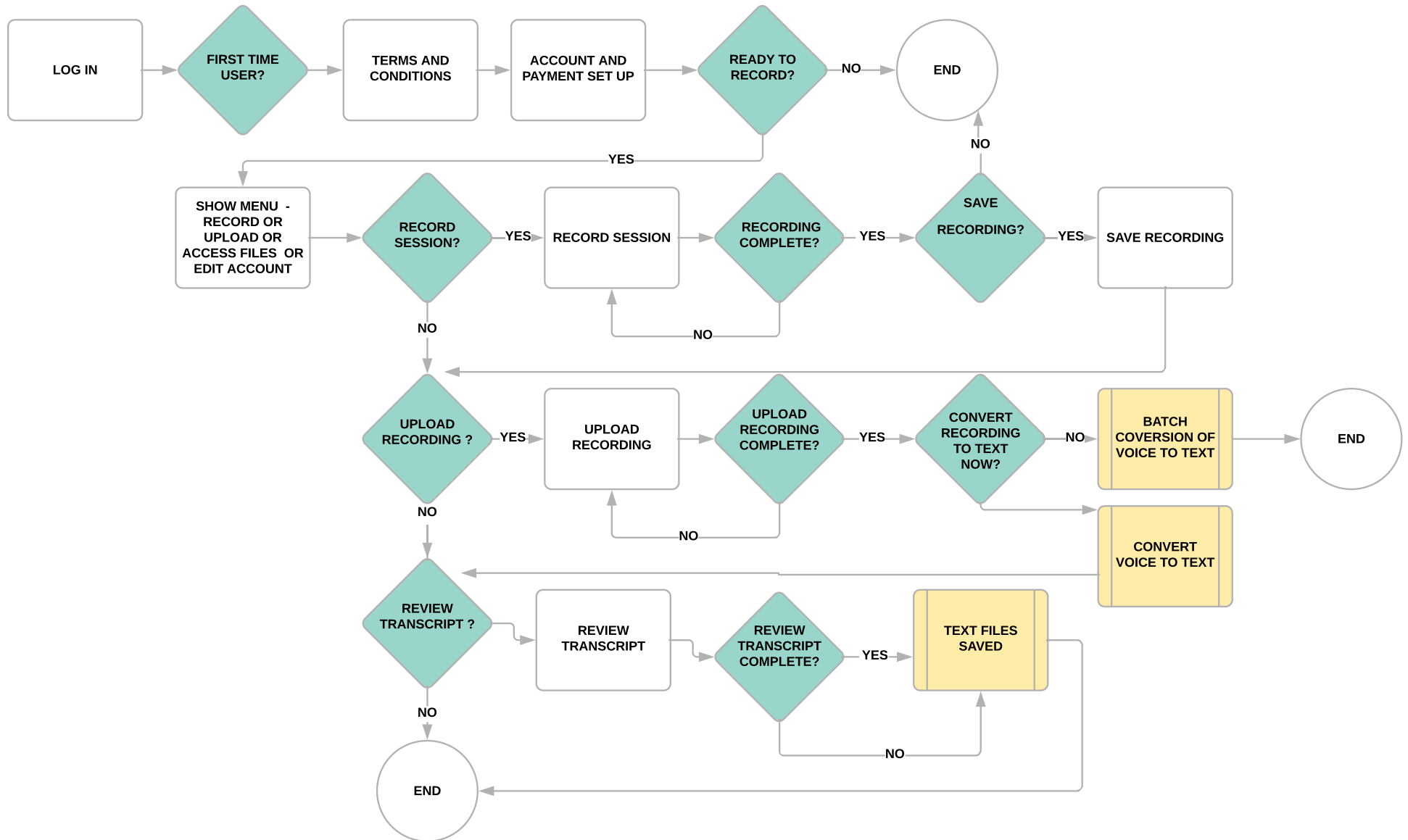
<https://stackoverflow.com/questions/9893175/google-text-to-speech-api>

<https://www.forbes.com/sites/christieterrell/2017/03/13/the-top-5-security-functions-to-outsource/#43c2969a2a5a>

<https://softwareengineering.stackexchange.com/questions/77206/what-is-your-application-versioning-strategy>

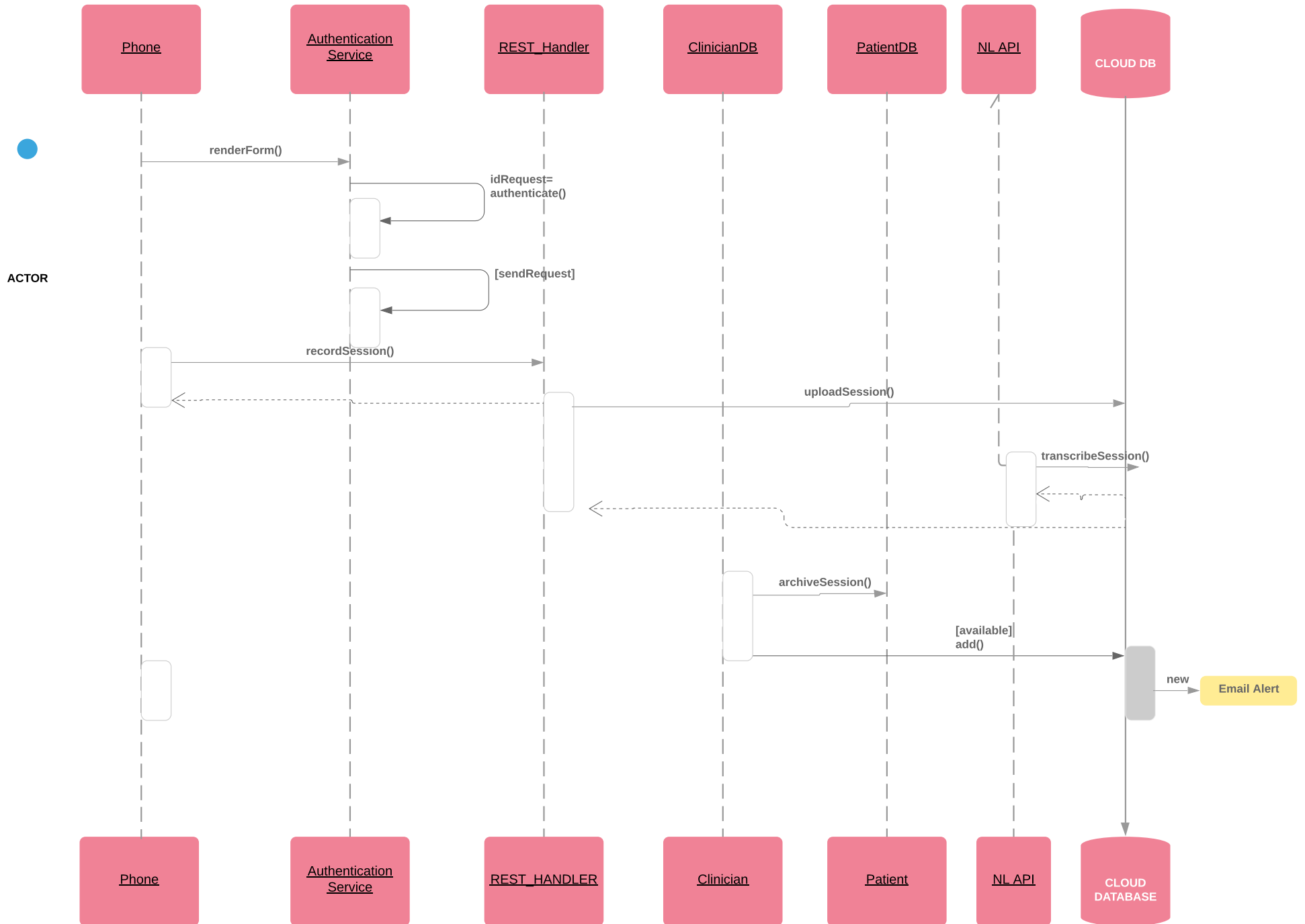
<https://www.softwaretestinghelp.com/>

https://www.capterra.com/sem-compare/mental-health-software?utm_source=ps-bing&utm_medium=cpc&gclid=CLHxk8DGnt8CFX2FxQldWSgGkA&gclidsrc=ds



HIGH LEVEL LGOICAL DATA MODEL FOR SANA





SANA DEPLOYMENT ARCHITECTURE

