CSE 237D: Digital Stethoscope Milestone Report

Amrita Moturi, Shayan Boghani, Shelby Myrman, Siya Rajpal

UC San Diego Computer Science & Engineering Dept.

MVP Progress

Proposed MVP

In our Project Specification Report, we laid out our initial vision for our original MVP below.

The goal of our Minimum Viable Product (MVP) is to develop a fully functional and portable digital stethoscope that captures, amplifies, records, and visualizes heart and lung sounds. This MVP serves as a proof-of-concept, providing the foundational system architecture for our broader vision while allowing us to gather early user feedback and iterate over time. Our emphasis is on incremental improvement: each core function is selected for high feasibility and clear alignment with the project's long-term clinical goals which can be decoupled and tested as we make progress.

Core MVP Components

1. **Acoustic Signal Acquisition:** The MVP will use a digital microphone with sensitivity in the appropriate frequency bands for heart and lung sounds. The system will be integrated into a robust enclosure modeled after existing stethoscope chest pieces to facilitate familiarity and ergonomics.

2. **Recording and Playback:** The MVP must reliably capture sound data for local storage and enable playback through either a speaker or through the front end software interface. Data will be stored securely

and labeled with timestamps, creating a foundation for potential clinical documentation or further diagnostic analysis.

3. **Portable and Durable Hardware Design:** The stethoscope will be encased in a 3D-printed shell that is both lightweight and resilient. It will house the microcontroller, power supply (battery), and microphone. The prototype will prioritize modularity to support future iterations, such as integration of custom PCBs or embedded ML components.

4. **Visualization via Companion Interface:** The MVP will include a simple user interface (UI) that visualizes the sound waveform and increases the sound captured by the device. This UI will be implemented either as a mobile app based on preliminary user research and will focus on clarity and usability. The visualization and audio allows real-time signal monitoring, enabling both users and potential clinical testers to assess stethoscope performance.

The MVP is intentionally narrow in scope but complete in functionality. Each feature is selected to highlight a critical part of the user experience and overall system integrity. The emphasis is not on completeness or polish but on creating an end-to-end experience that can be tested, validated, and iterated upon. By grounding the MVP in core biomedical functionality (amplification, recording, and visualization), we create a minimum usable product that is meaningful to stakeholders and can evolve based on feedback.

The MVP enables us to evaluate core assumptions about hardware compatibility, UI requirements, and clinical relevance before committing to more complex features such as headphone integration, HIPAA compliance, edge ML deployment, or custom PCB creation. It also provides an anchor for testing usability, sound quality, and data handling practices early in the development lifecycle. Ultimately, our MVP will demonstrate that a portable, secure, and intelligible digital stethoscope can serve as a launchpad for more advanced diagnostic tools and integration with health information systems.

MVP Progress Evaluation

The above four key components of our MVP can be individually evaluated to determine the team's progress.

1. **Acoustic Signal Acquisition:** The current device now uses a digital I2S microphone that has the proper frequency band to acquire heart and lung sounds. This microphone took some time to source, and as such testing was done with a more commonly available microphone. However, the microphone was swapped out and has been tested in the audio pipeline that was built out previously.

2. **Recording and Playback:** The audio pipeline has been rigorously tested to ensure robust quality throughout. The device is able to record and transmit information consistently and effectively for playback from our secure database. Each recording has metadata including: Device ID, Recording ID Number, and Reception Timestamp. These three identifiers allow for easy identification of each new recording.

3. **Portable and Durable Hardware Design:** This aspect of the device is currently a work in progress. The remainder of the quarter will be dedicated to creating this housing and testing it. This task will entail: designing the casing which allows the microphone to pick up key frequencies, ensuring this casing is robust (drop and water resistance), and finally integrating feedback into the casing design based on further user research.

4. **Visualization via Companion Interface:** This aspect of the product is currently a work in progress. We have developed the login, authentication, and most of the front-end of the new recording page. During week 7-8, we will pull device data from Firebase for the new recordings page and visualize the waveforms in the app. Additionally, while the data will be retrieved quickly, we will not monitor the signal in real-time, and there may be a lag due to our logistic and privacy-based decision to not upload from the device directly to a user's phone. We have accounted for this in our UX design to ensure the user experience still feels seamless.

Milestone and Deliverable Progress

Below, we lay out the milestones and deliverables proposed in our initial report and display the progress made as of this report.

Hardware

- **Deliverable 1 (Week 3): Hardware Schematic Diagram (Amrita)**
  Proof: Document with all major hardware components mapped out
  Milestones:

  - Determine and list major components needed based on final product requirements
  - Determine connections and additional interfacing components

  Progress: Here is the BOM regarding the critical components related to the hardware. This has been updated since Week 3 to reflect iterations in the hardware. Here is the Hardware Schematic Diagram which shows how the different components must be mapped/connected.

- **Deliverable 2 (Week 3): Hardware Selection Decision (Shayan)**
  Proof: Bill of materials based on hardware schematic diagram
  Milestones:

  - Research and downselect microcontrollers
  - Test microphones for heart and lung sound frequencies
  - Decide on filtering approach (analog and/or digital) to isolate target frequency bands for heart (typically 20–150 Hz) and lung (100–1000 Hz) sounds

  Progress: As referenced above, the BOM also served as the list of major components needed. Here is the BOM which includes the chosen microcontroller and microphone with the proper frequency range. This document contains details in how the hardware was chosen.

  - Hardware Selection ADR: Explains our decision to use a microcontroller for our system
  - MCU ADR: justifies our use of the ESP32 S3
  - Microphone Selection ADR: provides context for use of MAX4466 (analog) and INMP441 (digital) mics
  - Board Transition ADR: describes our initial transition to ESP-WROOM-32 board

- **Deliverable 3 (Week 5): Initial Product Prototype (Shayan)**
  Proof: Prototype with fully integrated hardware components
  Milestones:

  – Order and assemble hardware

  – Wire components, create documentation and diagram

  Progress: The initial hardware prototype can be seen in the demo video here. A newer, more streamlined version can be seen here.

- **Deliverable 4 (Week 5): Initial Prototype Code (Amrita)**
  Proof: Run and test hardware code, graphs showing frequency response before and after filtering, video demonstration
  Milestones:

  – Microphone data acquisition and processing

  – Recording functionality

  – Implement digital filtering to isolate relevant sound frequencies

  – Transmit data from microcontroller to device

  Progress: The initial prototype code can be seen functioning in the demo video here. The source code can be found in the GitHub repository here.

Software

- **Deliverable 1 (Week 2): Tech Stack Decision (Shelby)**
  Proof: Tech stack list and rationale
  Milestones:

  – Compare options and finalize stack

  – Write setup guide

  Progress: We have finalized our software technology stack, completed the setup guide, and documented our decisions in a series of ADRs.

  – Setup Guide: Explains how to run our demo code after downloading the repository.

  - Firebase ADR: justifies our use of Firebase Authentication and Firestore for backend data storage and user management
  - SwiftUI ADR: describes our decision to use Swift and SwiftUI based on user preference for native iOS apps
  - Python/FastAPI ADR: outlines why we chose Python with FastAPI for our backend services

- **Deliverable 2 (Week 2): GitHub Setup (Siya)**
  Proof: Screenshot of repo and CI/CD pipeline
  Milestones:

  - Initialize repo
  - Add CI/CD for linting and testing

  Progress: We have created our repository and organized it. A screenshot of it can be seen Here. We chose to organize the repository by hardware code, backend software code, and our Xcode project, *digital_stethoscope*. We have also included an admin folder in which you can find our system diagrams, and our ADRs.

  We have also completed the CI/CD pipeline. Seen in this screenshot of our GitHub actions, there are four workflows. Two for our Python backend, and two for our Swift frontend. You can see that the tests are run for every pull request, or any push or merge to our main branch. This ensures that we maintain good code quality. Note that the Pytest action is failing, because we are yet to write tests for all compoents of our backend. Our workflows can be found in our repository, under .github/workflows.

- **Deliverable 3 (Week 4): UI Prototyping (Shelby)**
  Proof: Wireframes and architecture document
  Milestones:

  - Create Figma wireframes
  - Diagram architecture for modularity

  Progress: We have completed our high-fidelity wireframes and system architecture for both the MVP and future versions of the system. This

Miro board contains our full Reach system software diagram and MVP system diagram, including database schematics for both.

You can find the MVP wireframes on Figma at the following links:

- Login Flow
- New Recording Page

We also created wireframes for the complete reach system:

- Learn More Flow
- Home Page
- FAQs Page
- History Flow
- Patient Flow
- Account Page

- **Deliverable 4 (Week 4): Frontend and Backend Setup (Siya)**
  Proof: Video explanation of system
  Milestones:

  - Backend init, sample route, local DB
  - Frontend init and routing

  Progress: We have successfully set up both our backend and our frontend. Videos walking through the structure/set up of both are below:

  - Backend Setup Walkthrough
  - Frontend Setup Walkthrough

- **Deliverable 5 (Week 5): Deploying Test App (Shelby)**
  Proof: Picture of test app on iPhone
  Milestones:

  - Configure deployment and fix permission issues
  - Upload to Apple test platform and test usability

Progress: After further research, we found that deploying a test app via TestFlight requires an Apple Developer Account, which costs $99 per year. While we initially considered using TestFlight to demo our working app, we determined that using the iPhone simulator in Xcode serves our current needs without incurring additional costs. As a result, we decided not to proceed with TestFlight at this stage. For more details, see our No TestFlight ADR here.

- **Deliverable 6 (Week 5): Add Authentication (Siya)**
  Proof: Video of working auth flow
  Milestones:
  - Backend schema and session logic
  - Frontend login and API integration

  Progress: We have completed the authentication flow end to end. Upon the creation of an account, it verifies that the user does not already have an account and assigns the user a distinct authentication token. From there, when logging in, it verifies the user based on the information stored, and gathers user data from the database. See our authentication flow walk through here.

Future Steps

Milestones and Deliverables

Hardware

- **Deliverable 5 (Week 7): Design and build casing for initial prototype (Shayan)**
  Proof: Take pictures of the assembled product and test to ensure all functionality still works with casing added
  Milestones:
  - Design casing in CAD and add hardware components to ensure dimensioning and tolerancing
  - 3D print casing
  - Assemble product

  - Test product and ensure all functionality remains

- **Deliverable 6 (Week 7): Transmit data from hardware device to database (Amrita)**
  Proof: Document process and have test data samples that have come from hardware in database
  Milestones:
  - Set up communication from hardware device to external gateway
  - Work with software team to ensure proper routing of received data from gateway to database
  - Perform repeatability testing to ensure that data transmission is consistent and the quality of data does not degrade across trials

- **Deliverable 7 (Week 9): Refine prototype design (Shayan)**
  Proof: New CAD design for the product which has a focus on ruggedness and ergonomics
  Milestones:
  - Get feedback from medical professionals on design
  - Incorporate feedback to make the product more useable
  - Improve casing to be drop resistant and have some water resistance for key components

- **Deliverable 8 (Week 9): Create the minimum viable product (Amrita)**
  Proof: Minimum Viable Product in hand
  Milestones:
  - Reprint casing and assemble product
  - Test product to ensure all functionality maintains previous quality

- **Deliverable 9 (Week 10): Create documentation for product (Shayan and Amrita)**
  Proof: Documents related to product and its abilities and use cases
  Milestones:
  - Compile documentation from full product development

    - Write product documentation to improve useability and adoption
    - Write specifications sheet related to limitations and expected use cases including safety information

Software

- **Deliverable 7 (Week 8): UI Complete (Shelby)**
  Proof: App walkthrough video
  Milestones:
    - Build and route core pages
    - Write tests and handle edge cases
- **Deliverable 8 (Week 8): Integration Test + Tagging (Siya)**
  Proof: Screenshot of database with tags
  Milestones:
    - Backend fetching and tagging
    - Integrate simple waveform display for audio recordings
    - Connect live frontend-backend and verify E2E
- **Deliverable 9 (Week 10): Web Presence (Shelby)**
  Proof: Link to website
  Milestones:
    - Create and deploy webpage
- **Deliverable 10 (Week 10): Additional Testing (Siya)**
  Proof: Screenshot of Testing scripts passing on Github Actions
  Milestones:
    - Additional frontend and backend testing
    - End to End testing

Schedule and Prioritization

Key	Priority
P0	Critical
P1	Important
P2	Stretch

Hardware

Milestone/Deliverable	Task	Due Date	Priority
Milestone	CAD Design for Dimensioning	May 16th	P0
Milestone	3D Print Casing	May 16th	P0
Milestone	Assemble product	May 16th	P0
Milestone	Test product functionality	May 16th	P0
Deliverable	Design and build initial prototype	May 16th	P0
Milestone	Communicate from device	May 16th	P0
Milestone	Data route integration with software	May 16th	P0
Milestone	Repeatability testing	May 16th	P0
Deliverable	Data transmission to database	May 16th	P0
Milestone	User Feedback	May 30th	P1
Milestone	Incorporate feedback	May 30th	P1
Milestone	Rugged Casing Design	May 30th	P1
Deliverable	Refine prototype design	May 30th	P1
Milestone	Assemble refined product	May 30th	P1
Milestone	Test refined product	May 30th	P1
Deliverable	Create the minimum viable product	May 30th	P0
Milestone	Compile dev documentation	June 6th	P0
Milestone	Write product documentation	June 6th	P0
Milestone	Write spec sheet	June 6th	P0
Deliverable	Compile all product documentation	June 6th	P0

Software

Milestone/Deliverable	Task	Due Date	Priority
Milestone	Build and route core pages	May 23rd	P0
Milestone	Write tests & handle edge cases	May 23rd	P1
Deliverable	UI Complete	May 23rd	P0
Milestone	Backend fetching and tagging	May 23rd	P0
Milestone	Integrate simple waveform display	May 23rd	P0
Milestone	Connect frontend-backend + verify E2E	May 23rd	P0
Deliverable	Integration Test + Tagging	May 23rd	P0
Milestone	Create and deploy webpage	June 7th	P1
Deliverable	Webpresence	June 7th	P1
Milestone	Additional frontend + backend testing	June 7th	P1
Deliverable	Additional testing	June 7th	P1

Change in Deliverables Reasoning:

Software, Deliverable 10: As mentioned above, deploying our app to Test Flight requires an Apple Developer account, which costs $99. The iOS simulator built into Xcode is sufficient for UI/UX testing and displaying our software end to end. Instead, we will use that time to fine-tune our backend and frontend unit tests along with end-to-end testing to ensure perfect functionality, which addresses the feedback we received to improve our testing infrastructure.

Project Specification Feedback

The Project Specification had some key feedback which the team incorporated into our progress.

- **Building a complete system, from scratch, in 10 weeks is very ambitious. That is your biggest risk.**
  On the software side, we have specified a reasonably-scoped MVP (a login flow and recording display page) to prioritize device data-capture and proof-of-concept. This way, if we do not complete our stretch features, such as patient linking, we will still have a working product that meets core user needs of amplification and playback.

  Hardware progress has been strong and the remaining tasks fit well in our remaining time frame. We did however choose to develop segments of the hardware (microphone acquisition, recording trigger, WiFi connection, Firebase connection, simple Firebase upload) so that each could be demonstrated independently with its own test code. This was very helpful in the integration phase as each process was fully developed. The main task became ensuring the available resources were used properly and without processes disrupting each other.

- **Think carefully about how to put the system into parts and allow those parts to be demonstrated individually.**
  On the software side, we have demonstrated this in our system diagrams. We are developing our software in 4 separate phases, with the first being our MVP. We can also demonstrate components of our MVP individually in various different ways. Our git commit history provides an extremely detailed demonstration of our system. Additionally, our MVP can be split into two parts: the authentication flow and the recording flow, and these can be demonstrated separately.

  The hardware was decoupled into task development. Each task that needed to be accomplished had its own test code written. This code was isolated to the basic functionality of that task. Once this code was tested and proven to be robust, it was integrated into the full workflow. The different tasks we identified were as follows:

  - Connect to WiFi (link, ADR)
  - Read button press from GPIO port (link)
  - Read and display serial data from microphone on Serial Monitor in Arduino IDE using I2S protocol (link)
  - Use button press to read data from microphone and display results in Serial Monitor (link)
  - Access and store items on microSD card using SPI protocol (link)
  - Store sample WAV file on microSD card and check output by playing back audio from microSD card on laptop (link)
  - Connect to Firebase and transmit sample audio recording (pre-recorded) to Firebase Storage and compare WAV file on microSD card to file in bucket (link)
  - Enable PSRAM and use to store audio data in buffer (link, ADR)

By decoupling these tasks, we sped up development and were able to integrate modules more seamlessly than attempting to write a monolithic codebase.

- **Testing infrastructure could use more thought. How will you test the individual components? How will you test the entire system?**

  We will test individual software components using unit tests written in Python and Swift, depending on the module. Integration tests will ensure interoperability between modules, and end-to-end testing will validate the full pipeline from input to output.

  Hardware testing is demonstrable through the module development code. Each component of the system can be isolated and tested. Additionally, the full pipeline can be debugged through the Serial Monitor if needed.

  For full system testing, we'll simulate real-world scenarios and inputs, using both automated and manual tests. Testing is now accounted for in our project milestones, with dedicated time allocated to verify functionality at each phase.

- **Documentation process could be better articulated. What exactly will you put into the repo?**
  We have improved our documentation process, and have defined what will go in the repository. We have organized the repository so the frontend code, the backend code, and the hardware code each have their own directory. Our README contains a setup guide, which we will expand as our project grows. We have also created an admin drectory containing system diagrams and ADRs. This directory will aid in documenting each stage of the development process, and inform the choices we made along the way. Additionally, logs are kept for daily progress on hardware here. These coincide with the hardware ADRs written for major decisions.

Oral Presentation Feedback

A few key updates have been made given this feedback:

- **Think more broadly about motivation. Maybe it could be used at home for people who don't want to have a stethoscope. That recording could also be done in remote visits with a doctor?**

  These comments have given us some ideas on how to shift our user research toward learning more about pain points regarding telehealth. We might also benefit from learning about how incorporating heart and lung sounds into health records might help medical professionals when evaluating patients in this context. People with high blood pressure often own devices at home to help them consistently monitor their readings. Similarly, a potential use case for our device would be to help those with pre-existing conditions monitor their heart and lung health without the need of a doctor visit. Rather than needing a complete stethoscope at home, our system could act as a device to record data and share it remotely with health providers.

- **Longer term doing some processing on device to analyze the heart would be valuable. That could be a simple as feedback on whether the signal is good**

  A stretch goal for the team is to perform onboard processing of the acquired audio data. Initially, our focus was to determine abnormalities that the patient needed to be made aware of immediately. However, it may be helpful to give the patient feedback on how they are operating the device (i.e. in telehealth use cases) to ensure they make adequate contact. This may be added as a long term goal due to time constraints.