CSC 394 – Autumn 2018

Project LIFE

Project Documentation

Table of Contents

Vision Statement

For patients and health service providers (HSPs) who want a more secure method of recording, storing, and accessing patient data, the LIFE Project is a software solution utilizing BlockChain processes which focuses on the security and access of the patient's records. Unlike modern HSP tools, LIFE Project is a simple, clean, and modern interface that is accessible for hospital workers, patients, and necessary third parties, such as insurance adjusters.

Section 1 – Project Overview

1.1 – Purpose and Scope

LIFE Project is being developed as a Capstone Project for the team members as a part of CSC 394 during the Autumn 2018 Quarter at DePaul University. The given assignment for the project was to utilize BlockChain technology to develop a solution for the current problems facing patients and HSPs regarding patient records, particularly in regards to access, accountability, and security.

During the time frame allotted for this project, the team will research, design, and develop a working BlockChain in order to address the given issues. We will create a system which will allow user to securely access the system in order to create, store, and retrieve records utilizing the BlockChain which will be constructed for this purpose. Additionally, the team will ensure that data is properly encrypted and verified by additional nodes.

Owing to time limitations, LIFE Project will not necessarily include HSP verification, usage tracking, or features beyond establishment of the BlockChain, P2P implementation, and account/record security. These and other features may be implemented as time allows.

1.2 – Goals and Objectives

1.2.1 Goals

1. Establish a secure method for the entry, storage, and retrieval of patient records
2. Ensure account and record security through logins and encryptions
3. Provide data validation through P2P networking
4. Allow for accountability in records through BlockChain implementation

1.2.2 Objectives

1. Create a BlockChain for LIFE Project which will store patient records as encrypted XML files.
2. Build a reliable user interface which will enable secure access to records.
3. Enable patient access to their records, through PC or mobile devices.
4. Provide accessible documentation for users and administrators.

1.3 – Project Deliverables

The following items will be delivered to the instructor, Professor Clark Elliott, on or before 11/13/2018:

1. Project Documentation
2. Final Demonstration of the finalized project
3. Personal Assessments for each member of the team
4. Capstone Paper

1.4 – Assumptions and Constraints

1.4.1 Assumptions

1 Team members will have regular and reliable access to their computers and to shared data.

2 Servers will be stable and readily accessible throughout project development.

1.4.2 Constraints

1 Approval is required in order to deviate from the requirements of the project, such as by changing the file type utilized for the storage of data.

1.5 – Schedule Summary

09/11/2018 – Project started

09/15/2018 – Version 0.1 Running

09/18/2018 – Proof of Concept Demonstration

09/25/2018 – Development Team solidified and permanent roles assigned

10/12/2018 – Project Vision solidified and Peer-to-Peer System begins development

10/17/2018 – Version 0.2 Running

10/23/2018 – Development Platform switched to Ethereum[1]

10/28/2018 – Version 0.3 Running

11/01/2018 – Code Frozen at Version 0.3.1

11/02/2018 – System Testing Complete

11/06/2018 – Project Completion Deadline

1.6 – Success Criteria

1 LIFE Project runs as intended on the project deadline.

2 LIFE Project is able to utilize a BlockChain to add, store, and retrieve patient records

3 Records in the BlockChain are properly encrypted, and verified by nodes.

4 All members of the team are well-versed in each aspect of the project

1.7 – Definitions

1.7.1 BlockChain

A **BlockChain** is a growing list of records, called blocks, which are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data

[1] From 10/22/2018 – 10/25/2018, two systems were developed in parallel: the original (Java-based), and the new (Solidity-based), until it was established that the changeover could occur without complication.

1.7.2 HSP

HSP is an initialism for **H**ealthcare **S**ervice **P**rovider, denoted in this document to represent employees at hospitals, doctor's offices, and other facilities utilized by patients in regards to their health.

1.7.3 P2P

P2P is an abbreviation for a **P**eer-**to**-**P**eer network, which is a distributed application architecture that partitions tasks or workloads between "peers". Peers are equally privileged, equipotent participants in the application. In regards to LIFE Project, such peers are the various nodes which are a part of the BlockChain verification system.

1.8 – Evolution of the Project Plan

Following the implementation of each version of LIFE Project, the project plan will be updated to reflect an adjusted schedule of tasks for the next iteration. This will be based on time invested in the previous iteration, as well as the scope of the next project version.

Reassessment of the project scope will be considered as a part of each adjustment in order to allow for deadlines and available time investments by the team.

Section 2 – Startup Plan

2.1 – Team Organization

Role	Person	Responsibilities
Project Manager	Leonard Helding	1. Responsible for the whole project. 2. Guarantee that all other responsibilities are met, or will initiate reassignments. 3. Must solve people problems in the group. 4. Guarantee that meetings will be held, will be efficient, and will have agendas 5. Manage the deadlines 6. Manage differing development styles among group members. 7. Be the bottom line for some tough decisions about deadlines, personnel, and group structure. 8. Liaison to instructor
Implementation Manager	Mark Burghard	1. Ensure that the project will actually run on each of the days that deliverables are due. 2. Manage the development team. 3. Oversee the translation of the design into running code. 4. Work with the Designer on technical issues relating to the running project. 5. Responsible for the passing grade of every group member. 6. Be the bottom line on whether or not any particular code will make it into the official final project. 7. Be able to guarantee that any code shown in demonstration will run. 8. Manage the backup systems such that they are ready to come online on a moment's notice during demonstrations. 9. Communicate with the Project Manager, the Designer, and the Planner, among others, about how the development is proceeding.

Planner	Laurel O’Such	1. Responsible for the plan that the whole group must follow this quarter. 2. Must create the formal, online planning document with dates, hours, modules, tasks, and names of group members responsible for each task, and the dependencies of one task on another task within it. 3. Work with the Project Manager. 4. Communicate regularly to report on the status of the plan 5. Provide positive feedback to group members by keeping everyone informed of the progress of all aspects of the group project 6. Prepare the final plan for the project booklet. Works with the Documentation manager. 7. Work with the Presentation Manager to have the plan ready for the RDP demo. 8. Make updates to the plan as tasks are completed through the quarter. 9. Work with the group and the Project Manager when the plan must be dynamically altered.
Requirements Manager	Brian Ivy	1. Responsible for the extensive formal requirements for the project. 2. Prepare the formal requirements document 3. Prepare the Requirements section of the RDP demo 4. Work with the Documentation Manager to prepare the requirements document for inclusion in the project booklet

Designer	Mark Burghard	1. Responsible for producing the full, technical design from which the project code will be written. 2. Work with the group to turn the initial ideas into a technical document that the group can follow. 3. Work with the Implementation Manager to select the hardware, database, client, server, and implementation platforms. 4. Perform the due diligence research to see what has been done before, what is available, what can be improved, how the design will solve problems for which there is a market. 5. Produce the formal design document. 6. Work with the Presentation Manager to prepare the design for the RDP demo.
Testing Manager	Laurel O’Such	1. Responsible for the testing plan for the project 2. Mange the testing efforts. 3. Produce formal tests of the project for the project booklet. 4. Oversee the design, plan, and execution of the testing for all requirements.
Presentation Manager	Logan Murray	1. Manage the critical, extensive demonstrations of the group's work 2. Have executive control over the look, and effectiveness, of the group during the critical minutes of the presentation. 3. Guarantee that the running project will be shown at the demo.

Webmaster	Adlai Abdelrazaq	1. Responsible for the central communication hub(s), and the "face" of the project group. 2. Work with the Implementation Manager when web-related services are needed for the group project. 3. Work with the Presentation Manager when web resources are required for the demonstrations. 4. Works with the Documentation Manager to translate the web resources into printed format for the project booklet.

2.2 – Project Communications

The team will utilize GroupMe for quick/immediate conversation and exchanges of information. Additionally, we will employee Google Drive for the purpose of sharing or uploading documents and files associated with the LIFE Project.

2.3 – Technical Process

For purposes of software development, team members will work within the determined language and framework(s) selected by the Implementation Manager for the finished project. Any additional portions of the coding or extraneous efforts, such as the website, may be done in other languages and made compatible with the main code with permission of the Implementation Manager.

Per the requirements of the project, the records stored in the block chain will be done in XML file format, unless otherwise approved by the instructor.

2.4 – Tools

- Programming Language – Solidity
- Version Control – Source code will be stored in a GitHub repository
- Build Tools – Local and main builds will be done using Ethereum
- Testing – Unit and individual tests will be implemented using Ganache

Appendix I: Project Charter

Project Title: LIFE	**Project Manager:** Leonard Helding
Start Date: 09/11/2018	**Project Sponsor:** Clark Elliott
End Date: 11/06/2018	**Customer:** DePaul University

Purpose:
To create a complete canonical record of medical treatment for patients and for providers.

Goals and Objectives:
We will create a peer-to-peer blockchain application that has the following properties:

- Once patient event records are entered into the system, the record becomes permanent, and cannot be altered.
- All records become part of a central ledger that is shared among all members of the multicast group.
- The marshaled external data format of the ledger is XML.
- All records have unique identifiers each of which (identifier) is electronically signed by the creating node.
- We will use three peers in our system, but the technology must be scalable to hundreds of peers.
- Create a web console interface for the input and display of patient event data.
- Create a web interface for reporting information from the ledger with at least the following features:
  - List all event records by date of event.
  - List all event records for a specific patient, by date of event
  - List all event records by provider by date of event
  - List all event records within a certain date window, by date of event.
  - List all prescriptions for a patient, with provider and by date

Schedule Information:

09/11/2018 – Project started
09/15/2018 – Version 0.1 Running
09/18/2018 – Proof of Concept Demonstration
09/25/2018 – Development Team solidified and permanent roles assigned
10/12/2018 – Project Vision solidified and Peer-to-Peer System begins development
10/17/2018 – Version 0.2 Running
10/23/2018 – Development Platform switched to Ethereum
10/28/2018 – Version 0.3 Running
11/01/2018 – Code Frozen at Version 0.3.1
11/02/2018 – System Testing Complete
11/06/2018 – Project Completion Deadline

Financial Information:
Not Applicable

Project Priorities and Degrees of Freedom:
Dates are not flexible concerning presentations and project completion.
Roles are expected to be flexible, enabling team members to take on responsibilities as they arise.

Approach:
Focus will initially be on the basic features of the project: initializing and encryption of a BlockChain, Peer-to-Peer Networking, and data retrieval. Following the implementations of these features, focus will expand to include user interface and account security.

Constraints:
Approval is required in order to deviate from the requirements of the project, such as by changing the file type utilized for the storage of data.

Assumptions:
Team members will have regular and reliable access to their computers and to shared data. Servers will be stable and readily accessible throughout project development.

Success Criteria:
The project will be considered a success if:

1. LIFE Project runs as intended on the project deadline.
2. LIFE Project is able to utilize a BlockChain to add, store, and retrieve patient records
3. Records in the BlockChain are properly encrypted, and verified by nodes.
4. All members of the team are well-versed in each aspect of the project

Scope:
At a minimum level, LIFE Project will:

1. Enable users to securely log in and out
2. Store, encrypt, and retrieve records
3. Run on the Project Completion Deadline

Risk and Obstacles:
As a diverse team with differing availabilities and obligations, LIFE Project will need to communicate effectively in order to maintain progress towards project completion. With little experience in a technological project of this scale, we have only estimates about the amount of time and effort which this will take in order to meet the deadlines and expectations which we have outlined in this document.

Appendix II: LIFE Project Management Plan

Team Members

Abdelrazaq, Adlai

Burghard, Mark

Helding, Leonard

Ivy, Brian

Murray, Logan

O'Such, Laurel

Tellis, Marvin

Document Control

Change History

Revision	Change Date	Description of Changes
V1.0	10/17/2018	Document Outlined
V1.1	10/30/2018	Space for Team Information added
V2.0	11/01/2018	New Template applied to document
V2.1	11/06/2018	Additional data added to all section.

Document Storage

This document is stored online in the team's shared Google Drive folder at:

https://drive.google.com/drive/u/0/folders/18hRBev6DbBU0UrHp14rXK4eyLMe6gC9d

Document Owner

Leonard Helding is responsible for developing and maintaining this document.

Appendix III: Team Biographies

III. A – Adlai Abdelrazaq

Adlai Abdelrazaq was the Webmaster for this project. He wrote the server-side code that handles the HTTP requests, and interacts with the Solidity contract for Ethereum. He also set up the MySQL databases and designed the schemas for the tables. Additionally, he wrote the frontend code that serves as the client for the application, and designed the interface. He was responsible for the server which hosted both the application and the group website on Apache web servers running on Ubuntu.

III.B – Mark Burghard

Mark Burghard is a Senior-year Computer Science major at DePaul university. Growing up, Mark found his passion in music, and to this day remains an avid musician, recording music and gigging around Chicago with his band, Harper's Grove. As the designer of this project, he brings his creativity and proficiency for abstract reasoning that he acquired through his musical experience to the project.

Mark discovered his second passion for computer science when he came to DePaul, and soon realized that he was well suited for code implementation as well as design. As such, he took on the role of implementation manager for the LIFE project and found himself enjoying the responsibilities of the position as well as the process of learning a new technology. Mark is experienced with a variety of object-oriented programming languages and is always up to the task of learning more as technology evolves.

III.C – Leonard Helding

Leonard is a Senior at DePaul University in Chicago. Though originally, he attended Columbia College to major in Film Directing, his life took an unexpected turn when his father passed away. After taking several years off from school in order to sort everything out, Leonard determined that where his passion lay was actually Computer Science.

A self-taught programmer prior to college, Leonard has eagerly delved into his CS degree and discovered a strong affinity for databases, project oversight, and communication. He hopes that these skills, combined with over a decade of management and office work experience will set him up to succeed in whatever comes next.

III.D – Brian Ivy

Brian Ivy is from Seattle Washington where he was born and raised. Brian has been married for 1 year and has two cats and a one-year old puppy. After graduating high school, Brian became an electrician and after five years of experience, he took a job offer in Ocean Side, California where he worked for the DOD (Department of Defense) as an electrical technician on LCAC's (Landing Craft Air Cushion, the largest hovercrafts in the world). Brian lived and worked on Camp Pendleton for 2 years and loved his job. However, in November of 2012 he was injured while working on an LCAC. Brian was unable to walk for six months and had to go to physical therapy for the next 2 years. The military doctors wouldn't clear him to ever return to work due to the fact that the work is very dangerous and you cannot be partially disabled. So, Brian decided to work towards a career change and he chose Computer Science. Brian is in his Senior year and about to finish up at DePaul. After he graduates, Brian would like to pursue a career as a system analyst working on ERP for a large company.

III.E – Logan Murray

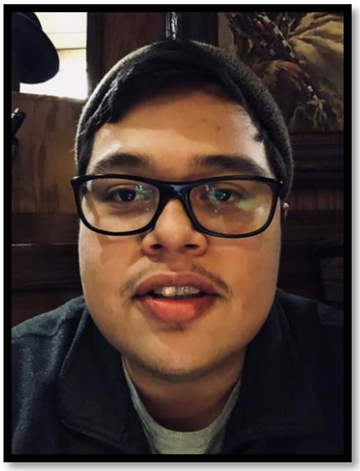
Technology and the complexity of computers has always been an interest of Logan's. To be raised in a world where technology is growing and advancing at such an exponential rate is exciting and a little overwhelming. Logan knew at a young age that we would soon be living in a world highly dependent on this advancing technology and he thought acquiring the knowledge and skill to work side-by-side it, rather from the outside looking in, was the smart thing to do.

Logan is the middle child between two sisters. He lives with his dad outside the city and commute to school every other day. Logan's time at DePaul has been challenging and rewarding, and he hopes that if he keeps up the hard work, he will eventually be prepared for the real world where his adventure truly begins. This project is a perfect example of the tenacity and strength that only DePaul students can bring to the table and Logan is proud of himself and his team members for the great work that was done.

III.F – Laurel O’Such

Laurel O’Such is a Senior, here at DePaul University. She is studying Computer Science with a concentration in Software Engineering and a minor in Digital Cinema. She is originally from Los Altos Hills, right in the heart of Silicon Valley. Growing up during the tech boom most likely played a role in why Laurel decided to study Computer Science, however she had never programmed before entering college and decided one day in winter quarter of freshman year that Computer Science was going to be her major. She hopes to one day move back to Northern California when the cost of living drops significantly, but has decided that her first move after college will be to New York City.

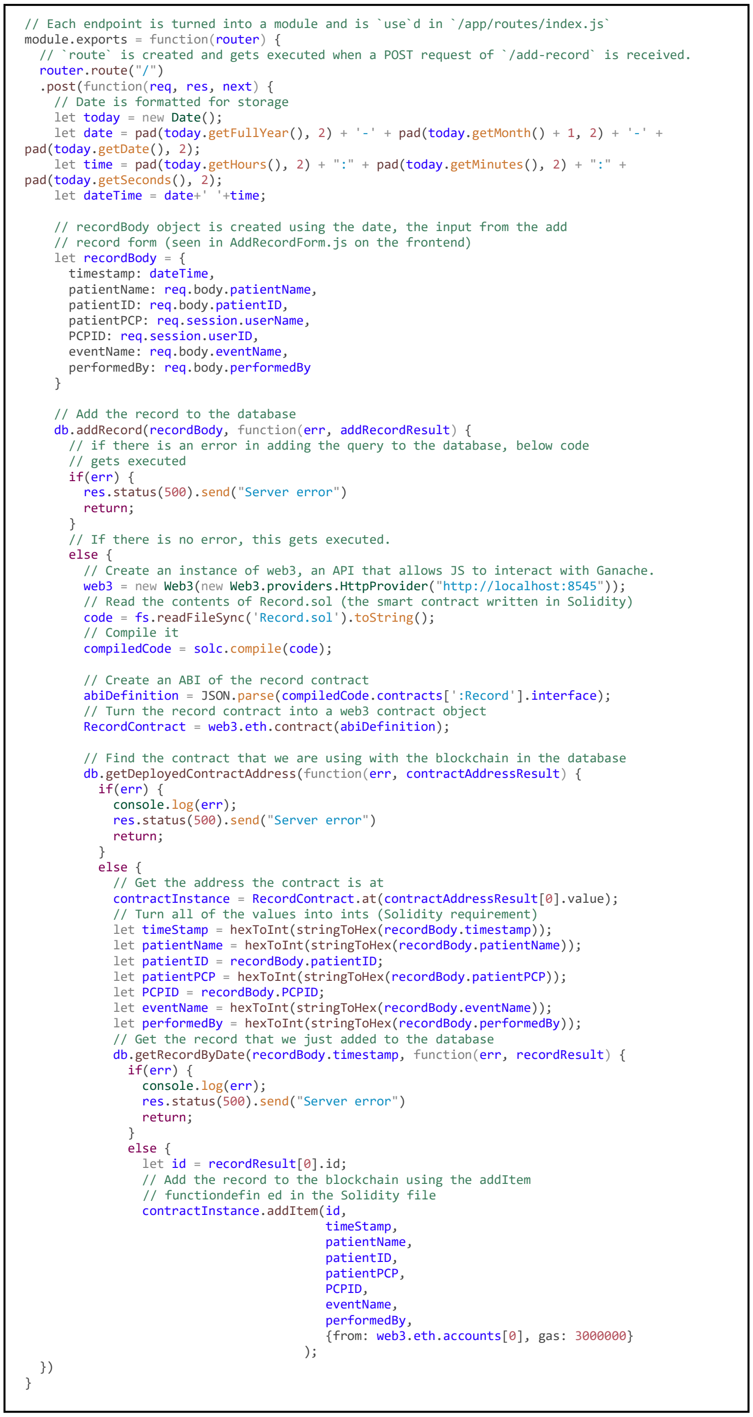
III.G – Marvin Telis



Marvin Telis is a Senior at DePaul University majoring in Computer Science with a concentration in Software Development. While planning to study Architecture in college right out of high school, he realized his true passion was in solving abstract problems and with the help of a few Google searches stumbled upon his major. Marvin was born and raised in Chicago and while attending a public high school in the city, he quickly realized the education system does not allow for its students to form an idea of what career paths exist once you graduate.

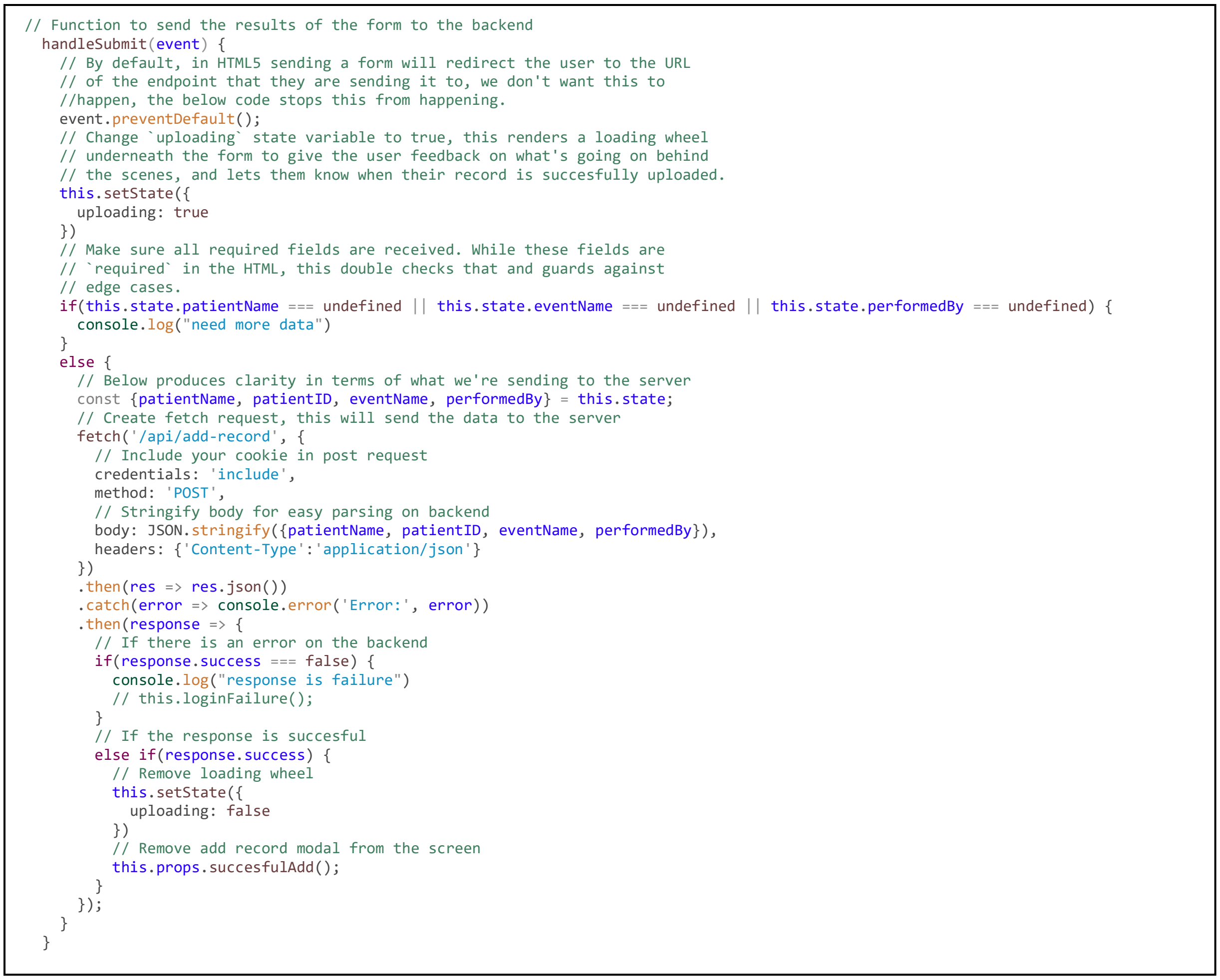
He hopes to use his knowledge in Computer Science to help the future generations educate themselves on the many career choices that exist in the real world. Marvin also hopes to develop mobile applications to help those in underprivileged neighborhoods in Chicago find the resources needed to thrive individually and as a community. His long-term goal is to fund a non-profit organization that teaches people to solve issues within their communities through programming.

Appendix IV – Code Samples



```
// Each endpoint is turned into a module and is `use`d in `/app/routes/index.js`
module.exports = function(router) {
  // `route` is created and gets executed when a POST request of `/add-record` is received.
  router.route("/")
  .post(function(req, res, next) {
    // Date is formatted for storage
    let today = new Date();
    let date = pad(today.getFullYear(), 2) + '-' + pad(today.getMonth() + 1, 2) + '-' +
pad(today.getDate(), 2);
    let time = pad(today.getHours(), 2) + ":" + pad(today.getMinutes(), 2) + ":" +
pad(today.getSeconds(), 2);
    let dateTime = date+' '+time;

    // recordBody object is created using the date, the input from the add
    // record form (seen in AddRecordForm.js on the frontend)
    let recordBody = {
      timestamp: dateTime,
      patientName: req.body.patientName,
      patientID: req.body.patientID,
      patientPCP: req.session.userName,
      PCPID: req.session.userID,
      eventName: req.body.eventName,
      performedBy: req.body.performedBy
    }

    // Add the record to the database
    db.addRecord(recordBody, function(err, addRecordResult) {
      // if there is an error in adding the query to the database, below code
      // gets executed
      if(err) {
        res.status(500).send("Server error")
        return;
      }
      // If there is no error, this gets executed.
      else {
        // Create an instance of web3, an API that allows JS to interact with Ganache.
        web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));
        // Read the contents of Record.sol (the smart contract written in Solidity)
        code = fs.readFileSync('Record.sol').toString();
        // Compile it
        compiledCode = solc.compile(code);

        // Create an ABI of the record contract
        abiDefinition = JSON.parse(compiledCode.contracts[':Record'].interface);
        // Turn the record contract into a web3 contract object
        RecordContract = web3.eth.contract(abiDefinition);

        // Find the contract that we are using with the blockchain in the database
        db.getDeployedContractAddress(function(err, contractAddressResult) {
          if(err) {
            console.log(err);
            res.status(500).send("Server error")
            return;
          }
          else {
            // Get the address the contract is at
            contractInstance = RecordContract.at(contractAddressResult[0].value);
            // Turn all of the values into ints (Solidity requirement)
            let timeStamp = hexToInt(stringToHex(recordBody.timestamp));
            let patientName = hexToInt(stringToHex(recordBody.patientName));
            let patientID = recordBody.patientID;
            let patientPCP = hexToInt(stringToHex(recordBody.patientPCP));
            let PCPID = recordBody.PCPID;
            let eventName = hexToInt(stringToHex(recordBody.eventName));
            let performedBy = hexToInt(stringToHex(recordBody.performedBy));
            // Get the record that we just added to the database
            db.getRecordByDate(recordBody.timestamp, function(err, recordResult) {
              if(err) {
                console.log(err);
                res.status(500).send("Server error")
                return;
              }
              else {
                let id = recordResult[0].id;
                // Add the record to the blockchain using the addItem
                // functiondefin ed in the Solidity file
                contractInstance.addItem(id,
                                         timeStamp,
                                         patientName,
                                         patientID,
                                         patientPCP,
                                         PCPID,
                                         eventName,
                                         performedBy,
                                         {from: web3.eth.accounts[0], gas: 3000000}
                                      );
  })
}
```

The code to the left is one snippet that we selected to showcase. This is a small portion of our addRecord.js file, which we utilize to add new records to the database.



```
// Function to send the results of the form to the backend
  handleSubmit(event) {
    // By default, in HTML5 sending a form will redirect the user to the URL
    // of the endpoint that they are sending it to, we don't want this to
    //happen, the below code stops this from happening.
    event.preventDefault();
    // Change `uploading` state variable to true, this renders a loading wheel
    // underneath the form to give the user feedback on what's going on behind
    // the scenes, and lets them know when their record is succesfully uploaded.
    this.setState({
      uploading: true
    })
    // Make sure all required fields are received. While these fields are
    // `required` in the HTML, this double checks that and guards against
    // edge cases.
    if(this.state.patientName === undefined || this.state.eventName === undefined || this.state.performedBy === undefined) {
      console.log("need more data")
    }
    else {
      // Below produces clarity in terms of what we're sending to the server
      const {patientName, patientID, eventName, performedBy} = this.state;
      // Create fetch request, this will send the data to the server
      fetch('/api/add-record', {
        // Include your cookie in post request
        credentials: 'include',
        method: 'POST',
        // Stringify body for easy parsing on backend
        body: JSON.stringify({patientName, patientID, eventName, performedBy}),
        headers: {'Content-Type':'application/json'}
      })
      .then(res => res.json())
      .catch(error => console.error('Error:', error))
      .then(response => {
        // If there is an error on the backend
        if(response.success === false) {
          console.log("response is failure")
          // this.loginFailure();
        }
        // If the response is succesful
        else if(response.success) {
          // Remove loading wheel
          this.setState({
            uploading: false
          })
          // Remove add record modal from the screen
          this.props.succesfulAdd();
        }
      });
    }
  }
```

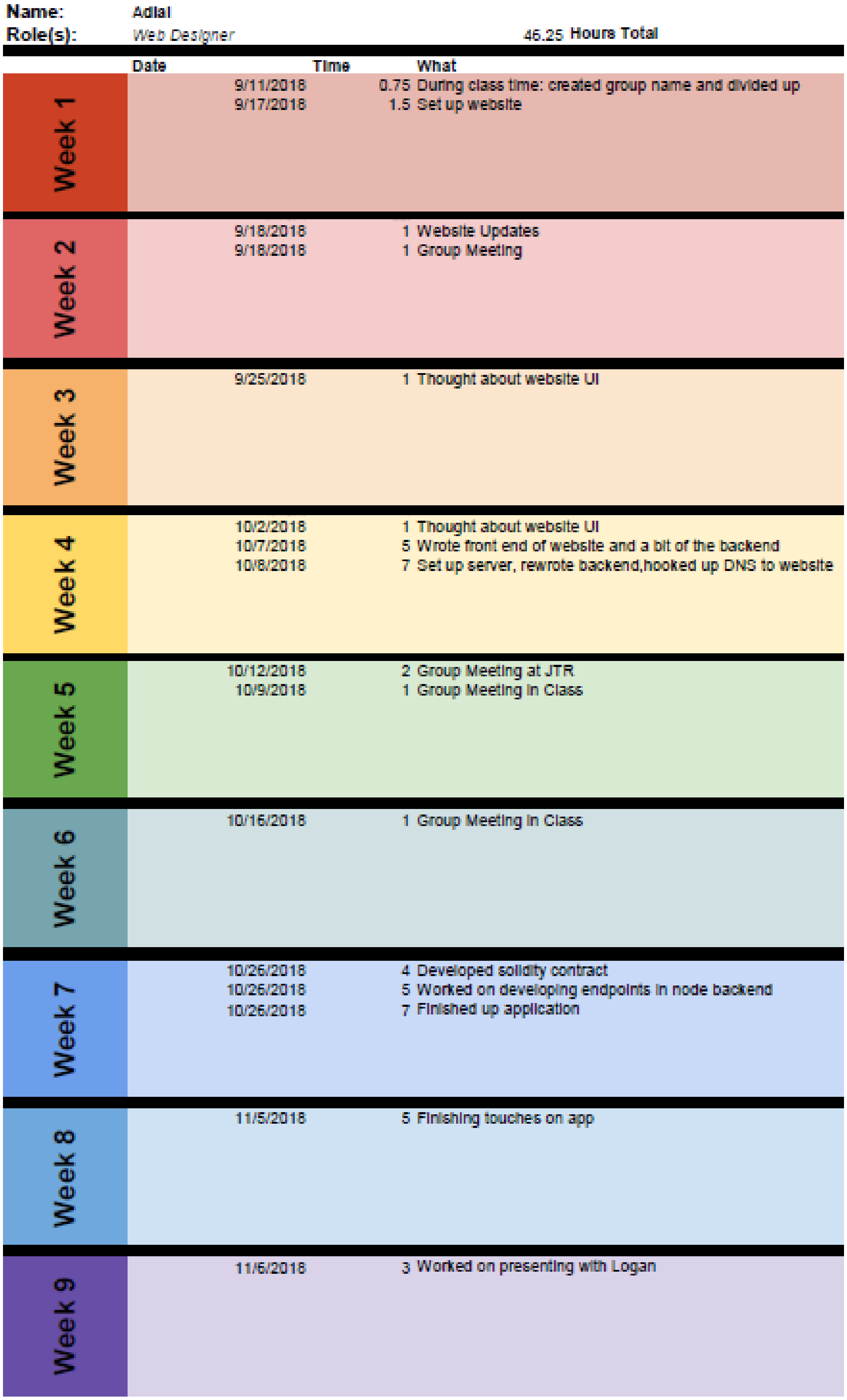
The above code is a sample from our addRecordForm.js, which is utilized by the client-interface on our website in order to enable the form to interact with the Solidity code on the back end.

Appendix V – Time Logs

V.A – Overview

Project LIFE											
249.2 Hours Invested											
Person	**Week 1**	**Week 2**	**Week 3**	**Week 4**	**Week 5**	**Week 6**	**Week 7**	**Week 8**	**Week 9**	**Week 10**	**Total**
Adlai	2.25	2	1	13	3	1	16	5	3	0	46.25
Brian	2.85	2.5	4.5	2.5	3	2	2.5	2	0	0	21.85
Laurel	2	4.5	1.5	4.5	5.75	2.75	3.75	4.85	1.75	0	31.35
Lee	7.3	8.55	7.5	5	5.7	5.35	5.55	4.9	7.75	0	57.6
Logan	4.25	4.5	2	5.5	4.5	2	3.5	2.75	1.75	0	30.75
Mark	5.75	7.4	4.5	8.5	3	4	6	0	0	0	39.15
Marvin	0.75	0	1.5	4.5	5	2.5	4	2.5	1.5	0	22.25
Total	**25.15**	**29.45**	**22.5**	**43.5**	**29.95**	**19.6**	**41.3**	**22**	**15.75**	**0**	**249.2**

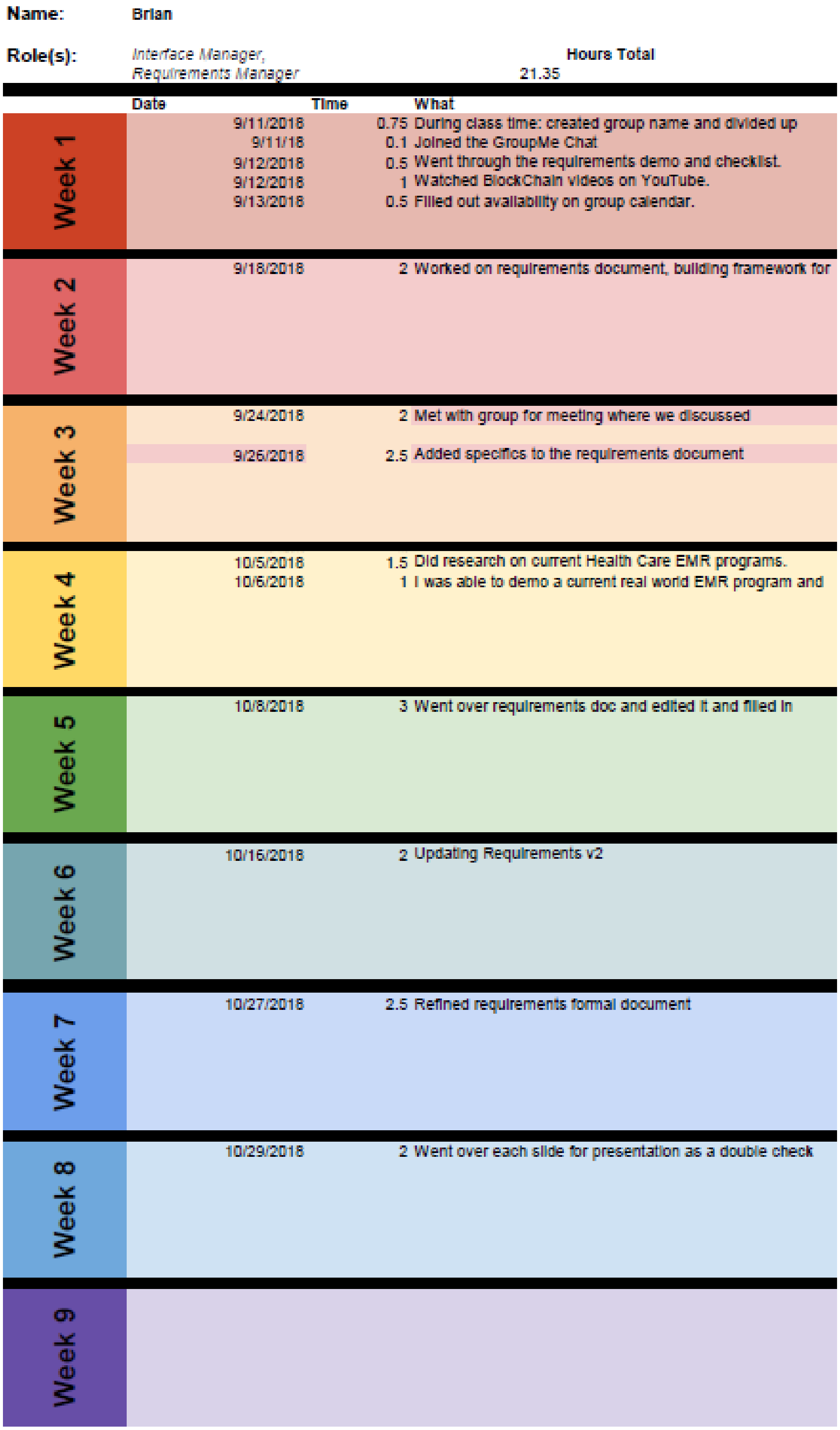
V.B – Individual Time Logs



Name: Adlai

Role(s): *Web Designer* 46.25 **Hours Total**

	Date	Time	What
Week 1	9/11/2018	0.75	During class time: created group name and divided up
	9/17/2018	1.5	Set up website
Week 2	9/18/2018	1	Website Updates
	9/18/2018	1	Group Meeting
Week 3	9/25/2018	1	Thought about website UI
Week 4	10/2/2018	1	Thought about website UI
	10/7/2018	5	Wrote front end of website and a bit of the backend
	10/8/2018	7	Set up server, rewrote backend,hooked up DNS to website
Week 5	10/12/2018	2	Group Meeting at JTR
	10/9/2018	1	Group Meeting In Class
Week 6	10/16/2018	1	Group Meeting In Class
Week 7	10/26/2018	4	Developed solidity contract
	10/26/2018	5	Worked on developing endpoints in node backend
	10/26/2018	7	Finished up application
Week 8	11/5/2018	5	Finishing touches on app
Week 9	11/6/2018	3	Worked on presenting with Logan

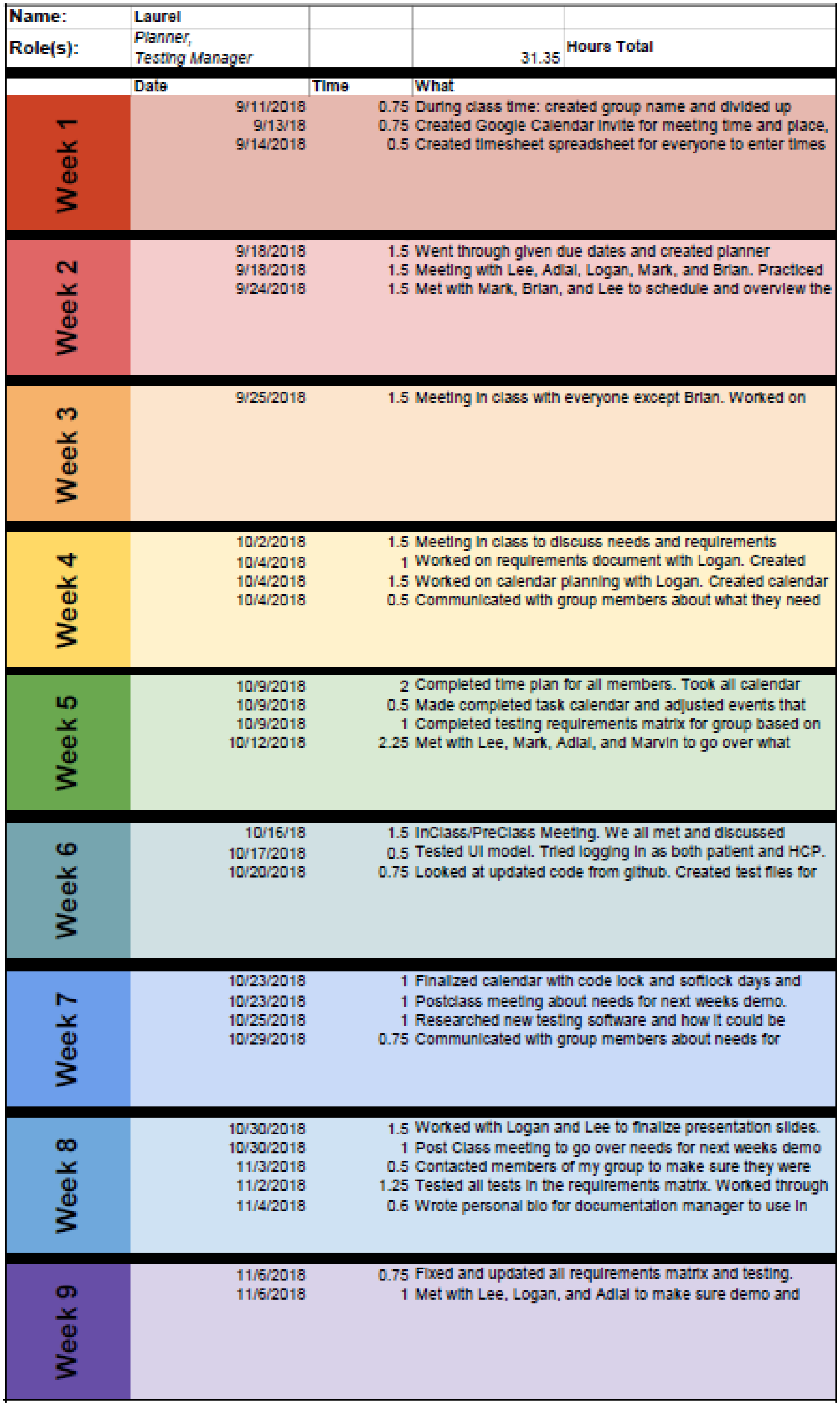


Name: Brian

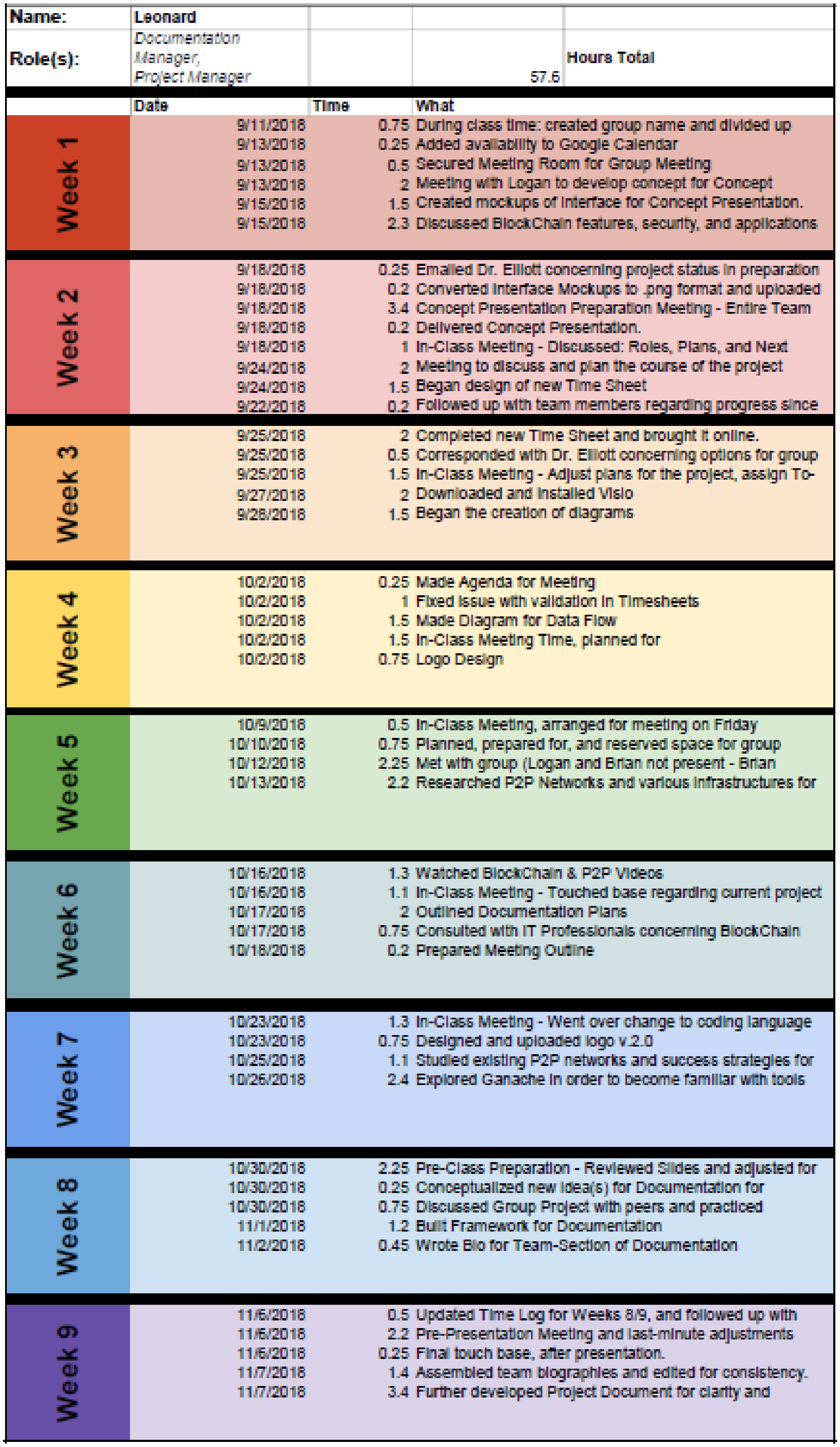
Role(s): *Interface Manager, Requirements Manager*

Hours Total 21.35

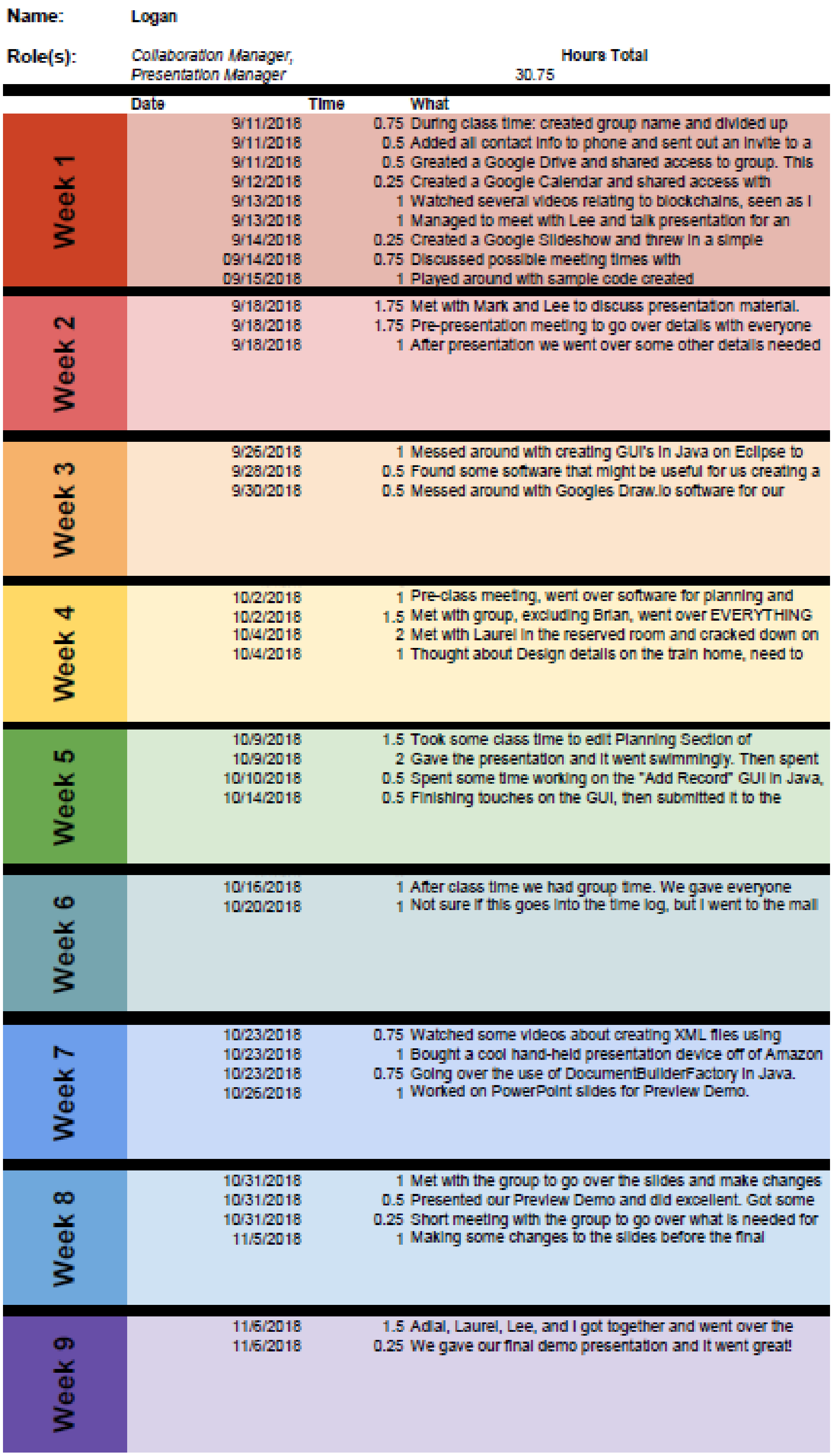
	Date	Time	What
Week 1	9/11/2018	0.75	During class time: created group name and divided up
	9/11/18	0.1	Joined the GroupMe Chat
	9/12/2018	0.5	Went through the requirements demo and checklist.
	9/12/2018	1	Watched BlockChain videos on YouTube.
	9/13/2018	0.5	Filled out availability on group calendar.
Week 2	9/18/2018	2	Worked on requirements document, building framework for
Week 3	9/24/2018	2	Met with group for meeting where we discussed
	9/26/2018	2.5	Added specifics to the requirements document
Week 4	10/5/2018	1.5	Did research on current Health Care EMR programs.
	10/6/2018	1	I was able to demo a current real world EMR program and
Week 5	10/8/2018	3	Went over requirements doc and edited it and filled in
Week 6	10/16/2018	2	Updating Requirements v2
Week 7	10/27/2018	2.5	Refined requirements formal document
Week 8	10/29/2018	2	Went over each slide for presentation as a double check
Week 9			



Name:	Laurel			
Role(s):	*Planner,* *Testing Manager*		31.35	Hours Total
	Date	Time	What	
Week 1	9/11/2018	0.75	During class time: created group name and divided up	
	9/13/18	0.75	Created Google Calendar invite for meeting time and place,	
	9/14/2018	0.5	Created timesheet spreadsheet for everyone to enter times	
Week 2	9/18/2018	1.5	Went through given due dates and created planner	
	9/18/2018	1.5	Meeting with Lee, Adial, Logan, Mark, and Brian. Practiced	
	9/24/2018	1.5	Met with Mark, Brian, and Lee to schedule and overview the	
Week 3	9/25/2018	1.5	Meeting in class with everyone except Brian. Worked on	
Week 4	10/2/2018	1.5	Meeting in class to discuss needs and requirements	
	10/4/2018	1	Worked on requirements document with Logan. Created	
	10/4/2018	1.5	Worked on calendar planning with Logan. Created calendar	
	10/4/2018	0.5	Communicated with group members about what they need	
Week 5	10/9/2018	2	Completed time plan for all members. Took all calendar	
	10/9/2018	0.5	Made completed task calendar and adjusted events that	
	10/9/2018	1	Completed testing requirements matrix for group based on	
	10/12/2018	2.25	Met with Lee, Mark, Adial, and Marvin to go over what	
Week 6	10/16/18	1.5	InClass/PreClass Meeting. We all met and discussed	
	10/17/2018	0.5	Tested UI model. Tried logging in as both patient and HCP.	
	10/20/2018	0.75	Looked at updated code from github. Created test files for	
Week 7	10/23/2018	1	Finalized calendar with code lock and softlock days and	
	10/23/2018	1	Postclass meeting about needs for next weeks demo.	
	10/25/2018	1	Researched new testing software and how it could be	
	10/29/2018	0.75	Communicated with group members about needs for	
Week 8	10/30/2018	1.5	Worked with Logan and Lee to finalize presentation slides.	
	10/30/2018	1	Post Class meeting to go over needs for next weeks demo	
	11/3/2018	0.5	Contacted members of my group to make sure they were	
	11/2/2018	1.25	Tested all tests in the requirements matrix. Worked through	
	11/4/2018	0.6	Wrote personal bio for documentation manager to use in	
Week 9	11/6/2018	0.75	Fixed and updated all requirements matrix and testing.	
	11/6/2018	1	Met with Lee, Logan, and Adial to make sure demo and	



Name:	Leonard			
Role(s):	*Documentation Manager, Project Manager*		57.6	Hours Total
	Date	Time	What	
Week 1	9/11/2018	0.75	During class time: created group name and divided up	
	9/13/2018	0.25	Added availability to Google Calendar	
	9/13/2018	0.5	Secured Meeting Room for Group Meeting	
	9/13/2018	2	Meeting with Logan to develop concept for Concept	
	9/15/2018	1.5	Created mockups of interface for Concept Presentation.	
	9/15/2018	2.3	Discussed BlockChain features, security, and applications	
Week 2	9/18/2018	0.25	Emailed Dr. Elliott concerning project status in preparation	
	9/18/2018	0.2	Converted Interface Mockups to .png format and uploaded	
	9/18/2018	3.4	Concept Presentation Preparation Meeting - Entire Team	
	9/18/2018	0.2	Delivered Concept Presentation.	
	9/18/2018	1	In-Class Meeting - Discussed: Roles, Plans, and Next	
	9/24/2018	2	Meeting to discuss and plan the course of the project	
	9/24/2018	1.5	Began design of new Time Sheet	
	9/22/2018	0.2	Followed up with team members regarding progress since	
Week 3	9/25/2018	2	Completed new Time Sheet and brought it online.	
	9/25/2018	0.5	Corresponded with Dr. Elliott concerning options for group	
	9/25/2018	1.5	In-Class Meeting - Adjust plans for the project, assign To-	
	9/27/2018	2	Downloaded and installed Visio	
	9/28/2018	1.5	Began the creation of diagrams	
Week 4	10/2/2018	0.25	Made Agenda for Meeting	
	10/2/2018	1	Fixed issue with validation in Timesheets	
	10/2/2018	1.5	Made Diagram for Data Flow	
	10/2/2018	1.5	In-Class Meeting Time, planned for	
	10/2/2018	0.75	Logo Design	
Week 5	10/9/2018	0.5	In-Class Meeting, arranged for meeting on Friday	
	10/10/2018	0.75	Planned, prepared for, and reserved space for group	
	10/12/2018	2.25	Met with group (Logan and Brian not present - Brian	
	10/13/2018	2.2	Researched P2P Networks and various infrastructures for	
Week 6	10/16/2018	1.3	Watched BlockChain & P2P Videos	
	10/16/2018	1.1	In-Class Meeting - Touched base regarding current project	
	10/17/2018	2	Outlined Documentation Plans	
	10/17/2018	0.75	Consulted with IT Professionals concerning BlockChain	
	10/18/2018	0.2	Prepared Meeting Outline	
Week 7	10/23/2018	1.3	In-Class Meeting - Went over change to coding language	
	10/23/2018	0.75	Designed and uploaded logo v.2.0	
	10/25/2018	1.1	Studied existing P2P networks and success strategies for	
	10/26/2018	2.4	Explored Ganache in order to become familiar with tools	
Week 8	10/30/2018	2.25	Pre-Class Preparation - Reviewed Slides and adjusted for	
	10/30/2018	0.25	Conceptualized new idea(s) for Documentation for	
	10/30/2018	0.75	Discussed Group Project with peers and practiced	
	11/1/2018	1.2	Built Framework for Documentation	
	11/2/2018	0.45	Wrote Bio for Team-Section of Documentation	
Week 9	11/6/2018	0.5	Updated Time Log for Weeks 8/9, and followed up with	
	11/6/2018	2.2	Pre-Presentation Meeting and last-minute adjustments	
	11/6/2018	0.25	Final touch base, after presentation.	
	11/7/2018	1.4	Assembled team biographies and edited for consistency.	
	11/7/2018	3.4	Further developed Project Document for clarity and	

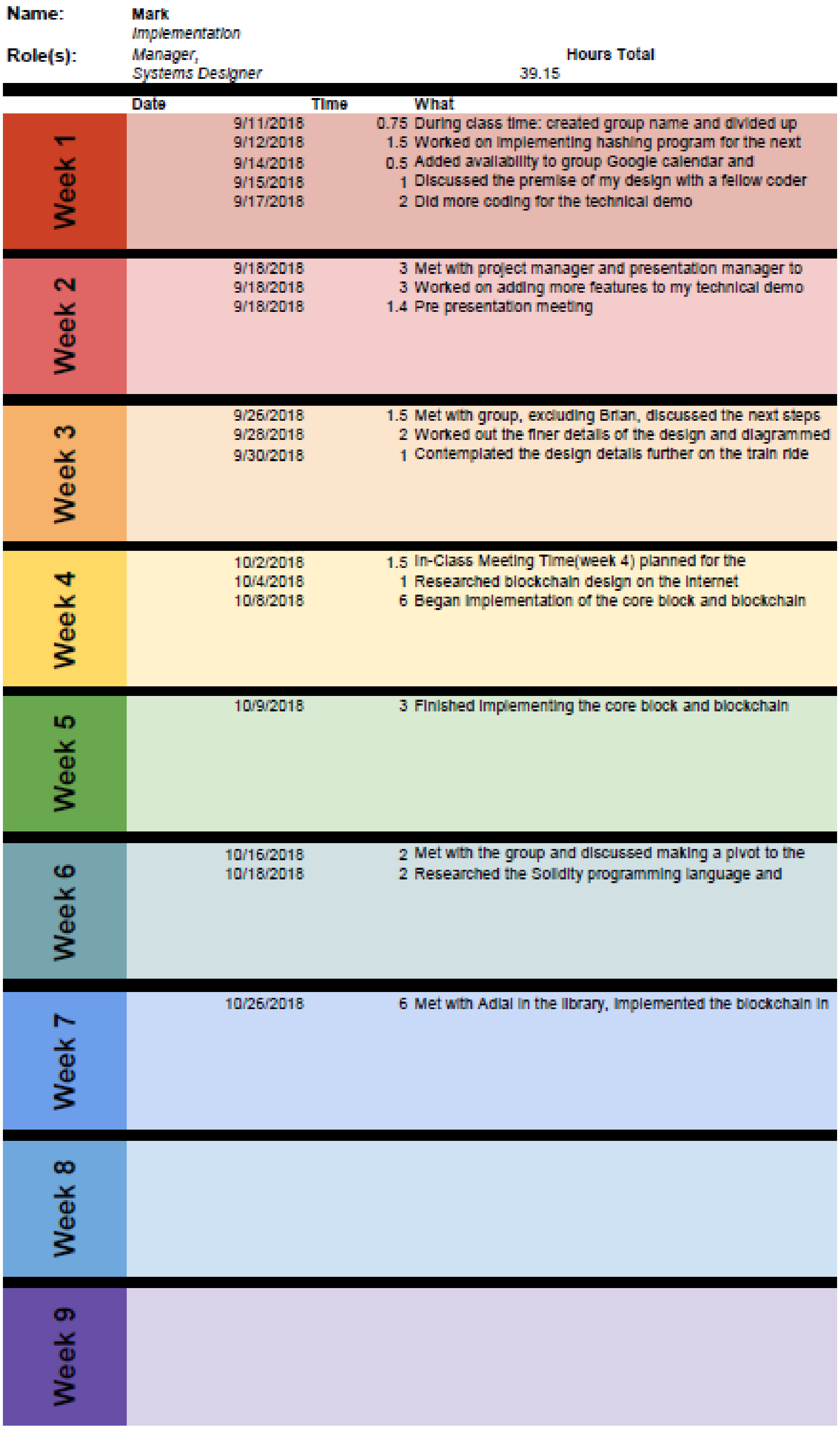


Name: Logan

Role(s): *Collaboration Manager, Presentation Manager*

Hours Total 30.75

	Date	Time	What
Week 1	9/11/2018	0.75	During class time: created group name and divided up
	9/11/2018	0.5	Added all contact info to phone and sent out an invite to a
	9/11/2018	0.5	Created a Google Drive and shared access to group. This
	9/12/2018	0.25	Created a Google Calendar and shared access with
	9/13/2018	1	Watched several videos relating to blockchains, seen as I
	9/13/2018	1	Managed to meet with Lee and talk presentation for an
	9/14/2018	0.25	Created a Google Slideshow and threw in a simple
	09/14/2018	0.75	Discussed possible meeting times with
	09/15/2018	1	Played around with sample code created
Week 2	9/18/2018	1.75	Met with Mark and Lee to discuss presentation material.
	9/18/2018	1.75	Pre-presentation meeting to go over details with everyone
	9/18/2018	1	After presentation we went over some other details needed
Week 3	9/26/2018	1	Messed around with creating GUI's in Java on Eclipse to
	9/28/2018	0.5	Found some software that might be useful for us creating a
	9/30/2018	0.5	Messed around with Googles Draw.io software for our
Week 4	10/2/2018	1	Pre-class meeting, went over software for planning and
	10/2/2018	1.5	Met with group, excluding Brian, went over EVERYTHING
	10/4/2018	2	Met with Laurel in the reserved room and cracked down on
	10/4/2018	1	Thought about Design details on the train home, need to
Week 5	10/9/2018	1.5	Took some class time to edit Planning Section of
	10/9/2018	2	Gave the presentation and it went swimmingly. Then spent
	10/10/2018	0.5	Spent some time working on the "Add Record" GUI in Java,
	10/14/2018	0.5	Finishing touches on the GUI, then submitted it to the
Week 6	10/16/2018	1	After class time we had group time. We gave everyone
	10/20/2018	1	Not sure if this goes into the time log, but I went to the mall
Week 7	10/23/2018	0.75	Watched some videos about creating XML files using
	10/23/2018	1	Bought a cool hand-held presentation device off of Amazon
	10/23/2018	0.75	Going over the use of DocumentBuilderFactory in Java.
	10/26/2018	1	Worked on PowerPoint slides for Preview Demo.
Week 8	10/31/2018	1	Met with the group to go over the slides and make changes
	10/31/2018	0.5	Presented our Preview Demo and did excellent. Got some
	10/31/2018	0.25	Short meeting with the group to go over what is needed for
	11/5/2018	1	Making some changes to the slides before the final
Week 9	11/6/2018	1.5	Adial, Laurel, Lee, and I got together and went over the
	11/6/2018	0.25	We gave our final demo presentation and it went great!

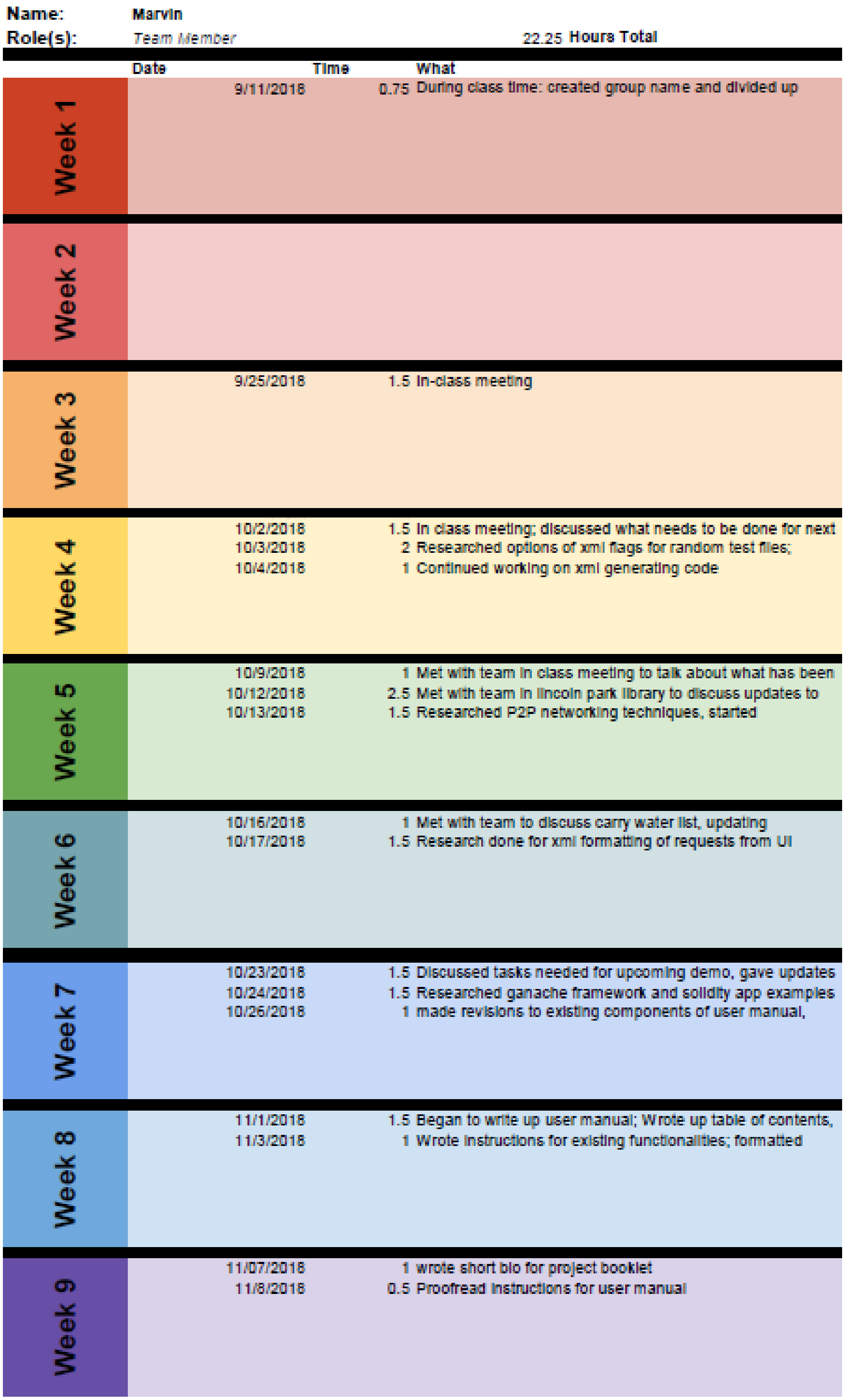


Name: Mark

Role(s): *Implementation Manager, Systems Designer*

Hours Total 39.15

	Date	Time	What
Week 1	9/11/2018	0.75	During class time: created group name and divided up
	9/12/2018	1.5	Worked on implementing hashing program for the next
	9/14/2018	0.5	Added availability to group Google calendar and
	9/15/2018	1	Discussed the premise of my design with a fellow coder
	9/17/2018	2	Did more coding for the technical demo
Week 2	9/18/2018	3	Met with project manager and presentation manager to
	9/18/2018	3	Worked on adding more features to my technical demo
	9/18/2018	1.4	Pre presentation meeting
Week 3	9/26/2018	1.5	Met with group, excluding Brian, discussed the next steps
	9/28/2018	2	Worked out the finer details of the design and diagrammed
	9/30/2018	1	Contemplated the design details further on the train ride
Week 4	10/2/2018	1.5	In-Class Meeting Time(week 4) planned for the
	10/4/2018	1	Researched blockchain design on the internet
	10/8/2018	6	Began implementation of the core block and blockchain
Week 5	10/9/2018	3	Finished implementing the core block and blockchain
Week 6	10/16/2018	2	Met with the group and discussed making a pivot to the
	10/18/2018	2	Researched the Solidity programming language and
Week 7	10/26/2018	6	Met with Adial in the library, implemented the blockchain in
Week 8			
Week 9			



Name: Marvin

Role(s): *Team Member*

22.25 **Hours Total**

	Date	Time	What
Week 1	9/11/2018	0.75	During class time: created group name and divided up
Week 2			
Week 3	9/25/2018	1.5	In-class meeting
Week 4	10/2/2018	1.5	In class meeting; discussed what needs to be done for next
	10/3/2018	2	Researched options of xml flags for random test files;
	10/4/2018	1	Continued working on xml generating code
Week 5	10/9/2018	1	Met with team in class meeting to talk about what has been
	10/12/2018	2.5	Met with team in lincoln park library to discuss updates to
	10/13/2018	1.5	Researched P2P networking techniques, started
Week 6	10/16/2018	1	Met with team to discuss carry water list, updating
	10/17/2018	1.5	Research done for xml formatting of requests from UI
Week 7	10/23/2018	1.5	Discussed tasks needed for upcoming demo, gave updates
	10/24/2018	1.5	Researched ganache framework and solidity app examples
	10/26/2018	1	made revisions to existing components of user manual,
Week 8	11/1/2018	1.5	Began to write up user manual; Wrote up table of contents,
	11/3/2018	1	Wrote instructions for existing functionalities; formatted
Week 9	11/07/2018	1	wrote short bio for project booklet
	11/8/2018	0.5	Proofread instructions for user manual

Appendix VI – Requirements Documentation

VI.A – Introduction

Purpose

The purpose of this document is to specify the requirements and preview some elements of the analysis model for Project LIFE.

LIFE is simple web-based software which consists of a user interface for health care providers and patients:

- **Health Care Provider Portal** - HCP's can chart on patients and record all of the necessary information
- **Patient Portal** – Patients can access their medical records and give access to new HCP's instantly which allows for telemedicine options.

Our Server: will store all of the electronic medical records in our database and makes use of color coding similar to paper charts so that the HCP experience is natural and familiar. The client computers will be able to connect to our server in order to retrieve the EMR. Patients can give access to HCP's, view, and print their records. Once a HCP has entered the data into the chart it will be entered into the block chain and enter the verification que. Once it has been verified it will become a part of the chain and can't be altered in any way providing a secure and safe record.

Document Conventions

Every requirement statement is assumed to have its own priority in-order to define in the most appropriate way, the system behavior. In addition, there are various figures that represent the described system, where it is needed, and serve only for better understanding of the implementation.

Intended Audience of this Document

This document is intended for any individual user, developer, tester, project manager or documentation writer that needs to understand the basic system architecture and its specifications.

Here are the potential uses for each one of the reader types:

- **The Developer**: The developer who wants to read, change, modify or add new requirements into the existing program, must firstly consult this document and update the requirements with appropriate manner to prevent the degradation of the original meaning of them and pass the information correctly to the next phases of the development process.
- **The User**: The user of this program reviews the diagrams and the specifications presented in this document and determines if the software has all the suitable requirements and if the software developer has implemented all of them.

- **Tester**: The tester needs this document to validate that the initial requirements of this programs actually corresponds to the executable program correctly.

For each one of the reader types to better understand this document, here is a suggestion of the chapters to read in this document:

- **Developer**: (The entire document)
- **User:** (Page #'s to follow)
- **Tester**: (Page #'s to follow)

Overview

1. Introduction: Provide an overview of the application, describe the document structure and point the individual objectives.

2. Overall Description: Provide the specification of the system model, the classes model, the main constraints and the list any assumed factors that used within this document.

3. System Features: Provide the analysis of the requirements by feature.

4. External Interface Requirements: Provide the visualization of the program and the requirements that are related with hardware, software and networking.

5. Other Nonfunctional Requirements: Provide some other constraints that apply to factors such as performance, safety, and security.

Project Scope

Our program offers an intuitive and natural interface for HCP to enter data in an EMR in the most secure way possible utilizing block chain technology. The patient will have access to their EMR and can allow other HCP access to their chart instantly which allows patients to utilize the benefits of telemedicine. The main objectives are to keep the EMR secure while allowing authorized parties instant access to the EMR which in turn, allows for faster and more efficient health care services while keeping the actual hospital lobbies and doctor offices from being overwhelmed at any given point. Our intentions are to streamline the entire health care service while keeping the data safe and secure.

VI.B – Overall Description

Product Perspective

This will look and feel as close to a paper chart as possible so that it feels natural for the HCP which will enable them to transition to an EMR. First and foremost, the HCP has to enjoy using it otherwise their could be resistance to switch to something new. We are building this with the HCP as our top priority as they will be using this program on an everyday basis where the patient will use it from time to time. There is a massive amount of data that a HCP may need to review and or enter, so their portal will be overwhelming to a patient who isn't familiar with the health care field so for that reason the patient portal will be stripped down and simplified as not to confuse the patient.

Due to the fact that most EMR programs that most HCP use is difficult to navigate without days of training where you learn specific knowledge of that program, our intention is to design a clean, intuitive, and friendly user interface where if you were to try to chart on a patient without any training you could figure it out rather quickly and easily and navigate through the various tabs and charts with ease.

Product Features

The major features this program contains are the following:

- **Cross platform support:** Offers operating support for most of the known and commercial operating systems
- **Language support:** Offers multiple language support for global use.
- **Statistic tool:** Query tool that can enable HCP to see trends and patterns in similar cases and case studies.
- **Flags:** Color coded flags to help identify vital data.
- **Printing:** Offers printing support for both the HCP and the patient.
- **HCP to HCP questions:** HCP will be able to ask other HCP questions on their medical advice and tap in to not just HCP that are within their office but their entire network.

User Classes and Characteristics

Physical Actors:

- **Health Care Provider:** The physician that is providing the patient with the service that they require.
- **Patient:** The patient will need access to their own EMR and allow other HCP to access their record instantly instead of waiting for the previous HCP to transfer the patients chart via snail mail or fax.

System Actors:

- **Client**: The client connects to the server and it handles the request based on the session login and finally submits the information back to the server.
- **Server**: The server is the system that accepts multiple connections from clients and enters the data into the database.

The primary actor is the HCP that connects to the server to enter the health care data.

Operating Environment

This program will operate in the following operating environment for the client and the server Graphical User Interface (GUI):

- Apple Mac OS X
- Linux/Unix
- Microsoft Windows

Design and Implementation Constraints

This program was created using Solidity programming language.

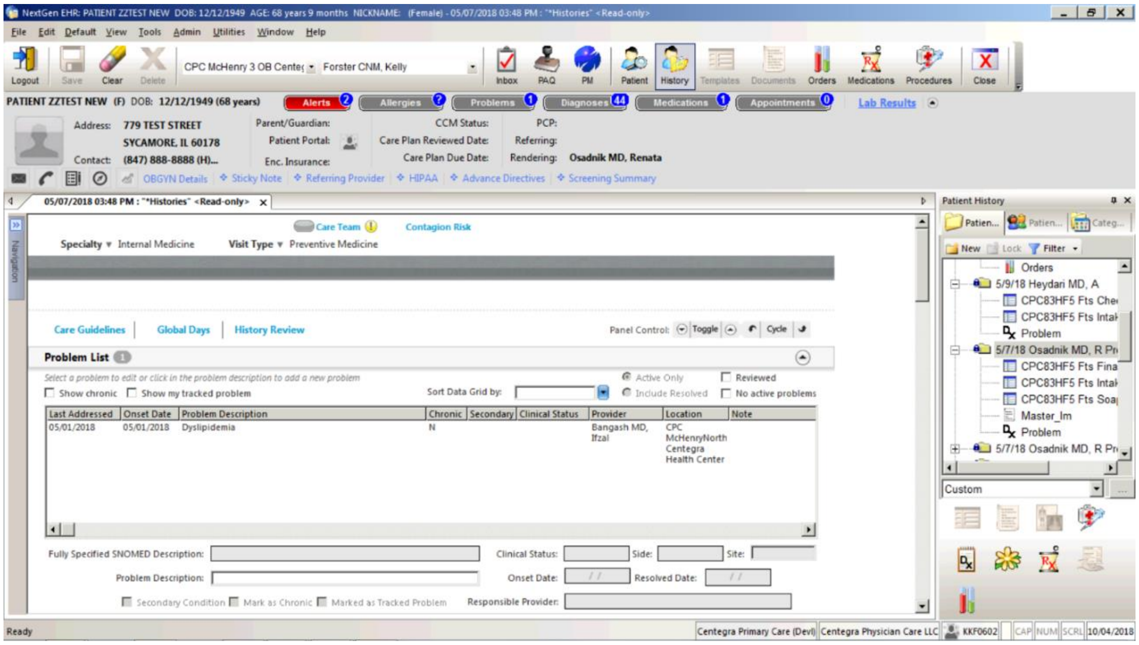
For language support except from the basic English language pack there is also many other language packs that can be enabled within the program.

Assumptions

So that the user can develop an understanding of the know-what to do rather than the know-how to do it we assume that the reader is not interested in knowing how to create and design windows forms and how the program was coded.

VI.C – System Features and Function

The HCP will begin the visit with vital signs, medications, allergies, and the patient's history.


NextGen EHR: PATIENT ZZTEST NEW DOB: 12/12/1949 AGE: 68 years 9 months NICKNAME: (Female) - 05/07/2018 03:48 PM : "*Histories" <Read-only>

CPC McHenry 3 OB Center · Forster CNM, Kelly

PATIENT ZZTEST NEW (F) DOB: 12/12/1949 (68 years)

Address: 779 TEST STREET
SYCAMORE, IL 60178
Contact: (847) 888-8888 (H)...

Rendering: Osadnik MD, Renata

05/07/2018 03:48 PM : "*Histories" <Read-only>

Specialty: Internal Medicine · Visit Type: Preventive Medicine

Problem List (1)

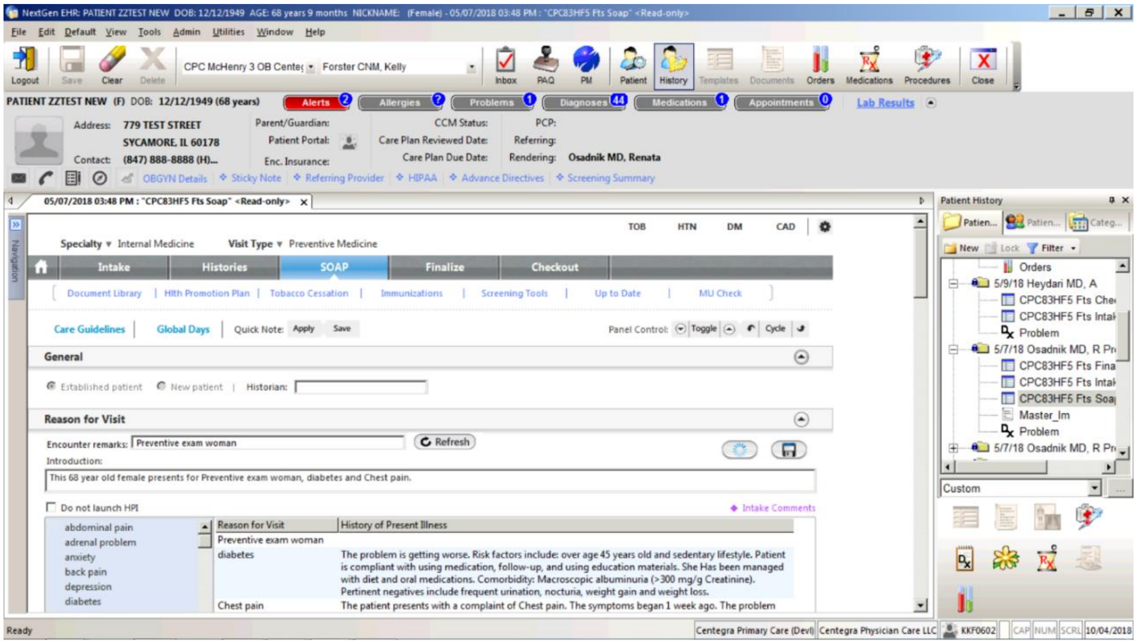
Select a problem to edit or click in the problem description to add a new problem

Last Addressed	Onset Date	Problem Description	Chronic	Secondary	Clinical Status	Provider	Location	Note
05/01/2018	05/01/2018	Dyslipidemia	N			Bangash MD, Ifzal	CPC McHenryNorth Centegra Health Center	

Ready · Centegra Primary Care (Devl) · Centegra Physician Care LLC · KKF0602 · 10/04/2018

Figure 1 - Example of Current HSP Software

The SOAP tab is where HCP's document the reason for visit, review of systems, physical exam, and assessment and plan. Often information such as vital signs or in office tests pull into this tab from other areas of the EMR.


NextGen EHR: PATIENT ZZTEST NEW DOB: 12/12/1949 AGE: 68 years 9 months NICKNAME: (Female) - 05/07/2018 03:48 PM : "CPC83HF5 Fts Soap" <Read-only>

CPC McHenry 3 OB Center · Forster CNM, Kelly

PATIENT ZZTEST NEW (F) DOB: 12/12/1949 (68 years)

Address: 779 TEST STREET
SYCAMORE, IL 60178
Contact: (847) 888-8888 (H)...

Rendering: Osadnik MD, Renata

05/07/2018 03:48 PM : "CPC83HF5 Fts Soap" <Read-only>

Specialty: Internal Medicine · Visit Type: Preventive Medicine

General

● Established patient ○ New patient | Historian:

Reason for Visit

Encounter remarks: Preventive exam woman

Introduction:
This 68 year old female presents for Preventive exam woman, diabetes and Chest pain.

☐ Do not launch HPI

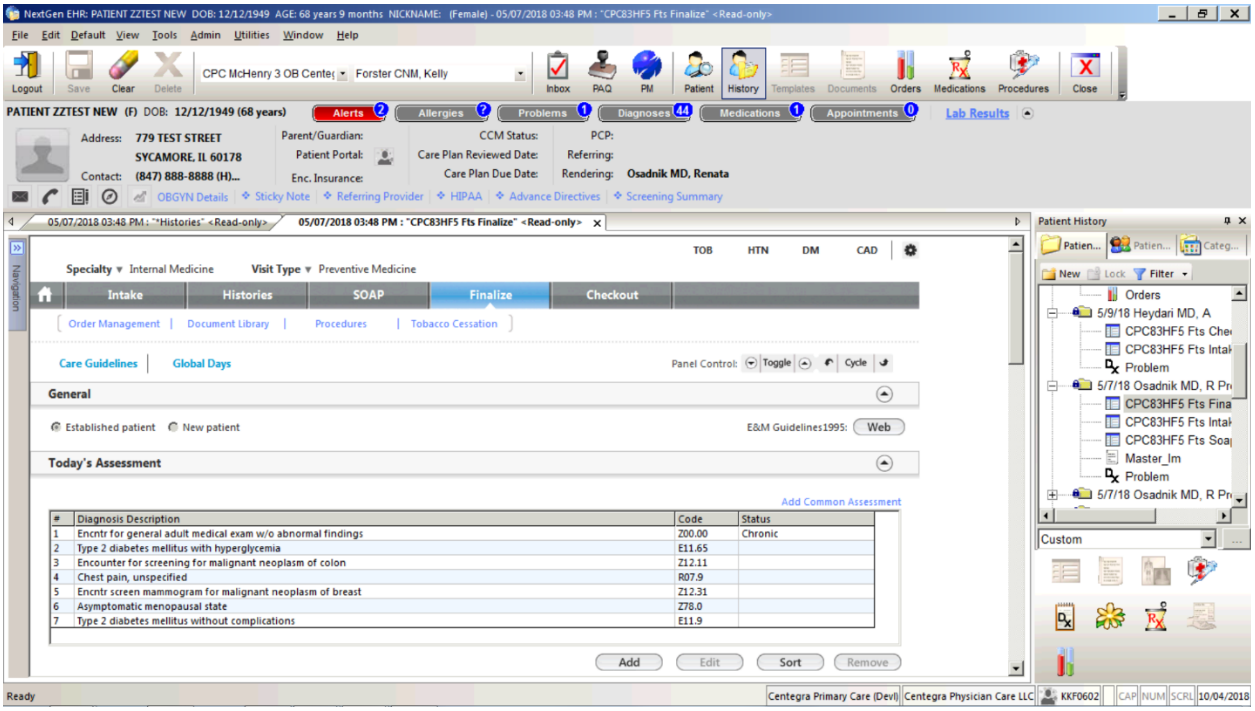
abdominal pain
adrenal problem
anxiety
back pain
depression
diabetes

Reason for Visit	History of Present Illness
Preventive exam woman	
diabetes	The problem is getting worse. Risk factors include: over age 45 years old and sedentary lifestyle. Patient is compliant with using medication, follow-up, and using education materials. She Has been managed with diet and oral medications. Comorbidity: Macroscopic albuminuria (>300 mg/g Creatinine). Pertinent negatives include frequent urination, nocturia, weight gain and weight loss.
Chest pain	The patient presents with a complaint of Chest pain. The symptoms began 1 week ago. The problem

Ready · Centegra Primary Care (Devl) · Centegra Physician Care LLC · KKF0602 · 10/04/2018

Figure 2 - Current Software SOAP Tab

This is the final part for coding and billing the visit.



NextGen EHR: PATIENT ZZTEST NEW DOB: 12/12/1949 AGE: 68 years 9 months NICKNAME: (Female) - 05/07/2018 03:48 PM : "CPC83HF5 Fts Finalize" <Read-only>

CPC McHenry 3 OB Cente; Forster CNM, Kelly

PATIENT ZZTEST NEW (F) DOB: 12/12/1949 (68 years)

Address: 779 TEST STREET
SYCAMORE, IL 60178
Contact: (847) 888-8888 (H)...

Parent/Guardian:
Patient Portal:
Enc. Insurance:

CCM Status:
Care Plan Reviewed Date:
Care Plan Due Date:

PCP:
Referring:
Rendering: Osadnik MD, Renata

05/07/2018 03:48 PM : "*Histories" <Read-only>
05/07/2018 03:48 PM : "CPC83HF5 Fts Finalize" <Read-only>

Specialty ▾ Internal Medicine   Visit Type ▾ Preventive Medicine

● Established patient ○ New patient   E&M Guidelines1995: Web

#	Diagnosis Description	Code	Status
1	Encntr for general adult medical exam w/o abnormal findings	Z00.00	Chronic
2	Type 2 diabetes mellitus with hyperglycemia	E11.65	
3	Encounter for screening for malignant neoplasm of colon	Z12.11	
4	Chest pain, unspecified	R07.9	
5	Encntr screen mammogram for malignant neoplasm of breast	Z12.31	
6	Asymptomatic menopausal state	Z78.0	
7	Type 2 diabetes mellitus without complications	E11.9	

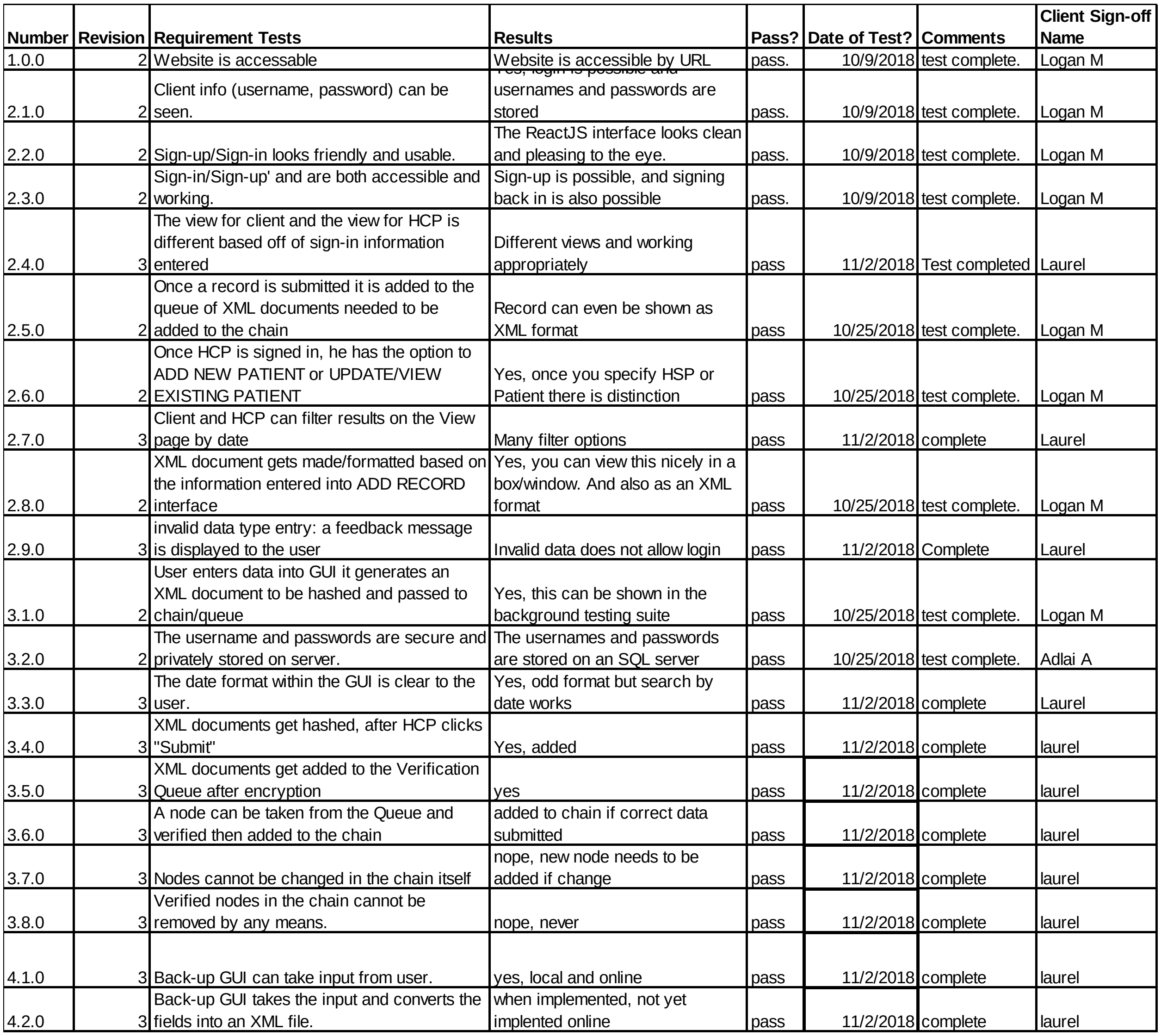
Centegra Primary Care (Devl) | Centegra Physician Care LLC | KKF0602 | 10/04/2018

Figure 3 - Current Software Billing

Appendix VII – Requirements Matrix

Requirement ID#	Revision	Requirement Description	Bad Client Attack	Bad Dev Attack	Approved?	Sign-Off Name
1.0.0	1	SQL server is up and running. Client info (username, password) can be seen.			Yes	Adlai
1.0.1	3	Quality check: User enjoys the look and feel of the entire program @ or above an 85% success rate based on user feedback.			Yes	Laurel
2.1.0	2	Username and Password text fields are displayed and usable for the client sign-in			Yes	Adlai
2.2.0	2	Sign-up/Sign-in looks friendly and usable to clients with basic computer skills/knowledge and overalll the client satifaction rating percentage is at 90% or more.			Yes	Adlai
2.2.1	2	Navigation to the Sign-in/Sign-up screen is easy and all clients have found and be directed to the screen within 5 seconds.			Yes	Adlai
2.3.0	2	The sign-in/sign-up is accessable through the website we've created.			Yes	Adlai
2.3.1	2	There are two options on the sign-in page, those are 'Sign-in/Sign-up' and are both accessible and working.			Yes	Adlai
2.4.0	2	Once client has signed in, the appropriate GUI is displayed			Yes	Adlai
2.4.1	4	Client and HCP can view previous medical records onced signed in.			Yes	Laurel
2.4.2	3	Once HCP is signed in, he has the option to ADD NEW PATIENT or UPDATE/VIEW EXISTING PATIENT			Yes	Laurel
2.5.0	4	At least 20 sample records are made and tested for early versions of the program.			Yes	Laurel
2.6.0	2	It is clear what entries need to be put into which text field on both the sign-in and GUI			Yes	Adlai
2.7.0	2	The client cannot add a record (only view), only the HCP can add and view.			Yes	Adlai
2.9.0	4	Fields only accept data types that the field requires. Testers will try to enter incorrect data types, a 100% success rate is required to pass.			Yes	Laurel
2.9.1	4	User feedback if invalid data type is entered into a field. Testers will check that in every invalid data type entry a feedback message is displayed to the user in red. Must have a 100% success rate.			Yes	Laurel
3.1.0	3	XML documents get hashed, after HCP clicks "Submit"			Yes	Laurel
3.1.1	2	After user enters data into GUI it generates an XML document to be hashed and passed to chain/queue at a 99% succes rate with a minimum of 100 submissions			Yes	Mark
3.1.2	2	Once a record is submitted it is added to the queue of XML documents needed to be added to the chain			Yes	Mark
3.2.0	2	The username and passwords are secure and privately stored on server.			Yes	Mark
3.3.0	2	The date format within the GUI is clear to the user.			Yes	Laurel
3.3.1	2	Client and HCP can filter results on the View page by date			Yes	Laurel
3.5.0	3	XML documents get added to the Verification Queue after encryption			Yes	Laurel
3.5.1	3	Sample records can be passed into Queue, verified, and added to chain without hiccups.			Yes	Laurel
3.6.0	2	Verification Queue is accessible and stored securely for those who would then be verifying the nodes.			Yes	Laurel
3.6.1	2	A node can be taken from the Queue and verified then added to the chain			Yes	Laurel
3.8.1	2	Verified nodes in the chain cannot be removed by any means.			Yes	Laurel

Appendix VIII – Testing Matrix



Number	Revision	Requirement Tests	Results	Pass?	Date of Test?	Comments	Client Sign-off Name
1.0.0	2	Website is accessable	Website is accessible by URL	pass.	10/9/2018	test complete.	Logan M
2.1.0	2	Client info (username, password) can be seen.	Yes, login is possible and usernames and passwords are stored	pass.	10/9/2018	test complete.	Logan M
2.2.0	2	Sign-up/Sign-in looks friendly and usable.	The ReactJS interface looks clean and pleasing to the eye.	pass.	10/9/2018	test complete.	Logan M
2.3.0	2	Sign-in/Sign-up' and are both accessible and working.	Sign-up is possible, and signing back in is also possible	pass.	10/9/2018	test complete.	Logan M
2.4.0	3	The view for client and the view for HCP is different based off of sign-in information entered	Different views and working appropriately	pass	11/2/2018	Test completed	Laurel
2.5.0	2	Once a record is submitted it is added to the queue of XML documents needed to be added to the chain	Record can even be shown as XML format	pass	10/25/2018	test complete.	Logan M
2.6.0	2	Once HCP is signed in, he has the option to ADD NEW PATIENT or UPDATE/VIEW EXISTING PATIENT	Yes, once you specify HSP or Patient there is distinction	pass	10/25/2018	test complete.	Logan M
2.7.0	3	Client and HCP can filter results on the View page by date	Many filter options	pass	11/2/2018	complete	Laurel
2.8.0	2	XML document gets made/formatted based on the information entered into ADD RECORD interface	Yes, you can view this nicely in a box/window. And also as an XML format	pass	10/25/2018	test complete.	Logan M
2.9.0	3	invalid data type entry: a feedback message is displayed to the user	Invalid data does not allow login	pass	11/2/2018	Complete	Laurel
3.1.0	2	User enters data into GUI it generates an XML document to be hashed and passed to chain/queue	Yes, this can be shown in the background testing suite	pass	10/25/2018	test complete.	Logan M
3.2.0	2	The username and passwords are secure and privately stored on server.	The usernames and passwords are stored on an SQL server	pass	10/25/2018	test complete.	Adlai A
3.3.0	3	The date format within the GUI is clear to the user.	Yes, odd format but search by date works	pass	11/2/2018	complete	Laurel
3.4.0	3	XML documents get hashed, after HCP clicks "Submit"	Yes, added	pass	11/2/2018	complete	laurel
3.5.0	3	XML documents get added to the Verification Queue after encryption	yes	pass	11/2/2018	complete	laurel
3.6.0	3	A node can be taken from the Queue and verified then added to the chain	added to chain if correct data submitted	pass	11/2/2018	complete	laurel
3.7.0	3	Nodes cannot be changed in the chain itself	nope, new node needs to be added if change	pass	11/2/2018	complete	laurel
3.8.0	3	Verified nodes in the chain cannot be removed by any means.	nope, never	pass	11/2/2018	complete	laurel
4.1.0	3	Back-up GUI can take input from user.	yes, local and online	pass	11/2/2018	complete	laurel
4.2.0	3	Back-up GUI takes the input and converts the fields into an XML file.	when implemented, not yet implented online	pass	11/2/2018	complete	laurel

Appendix IX – System Diagrams

Figure 4 - Login Diagram

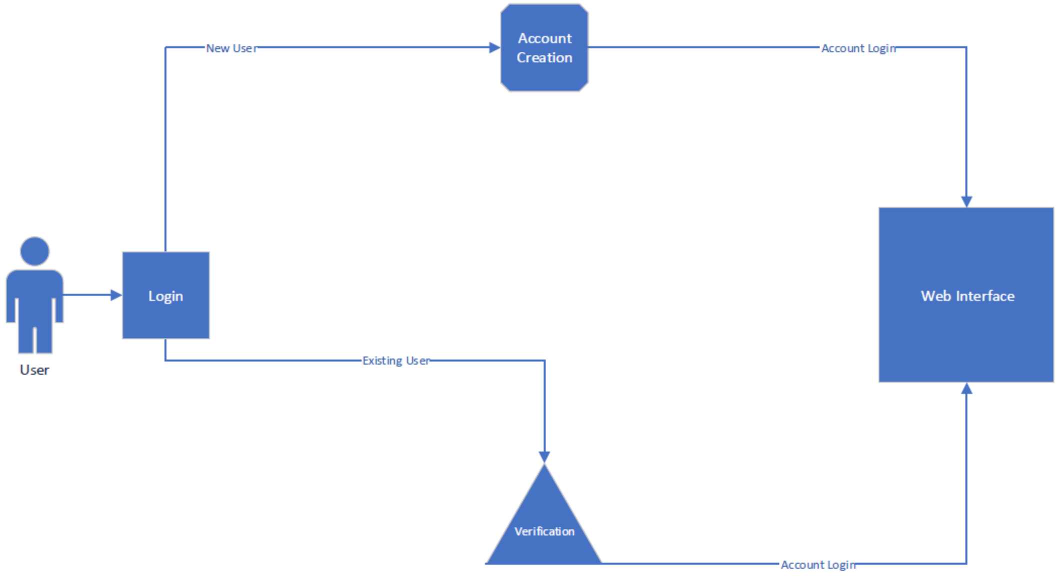


Figure 5 - Account Creation Diagram

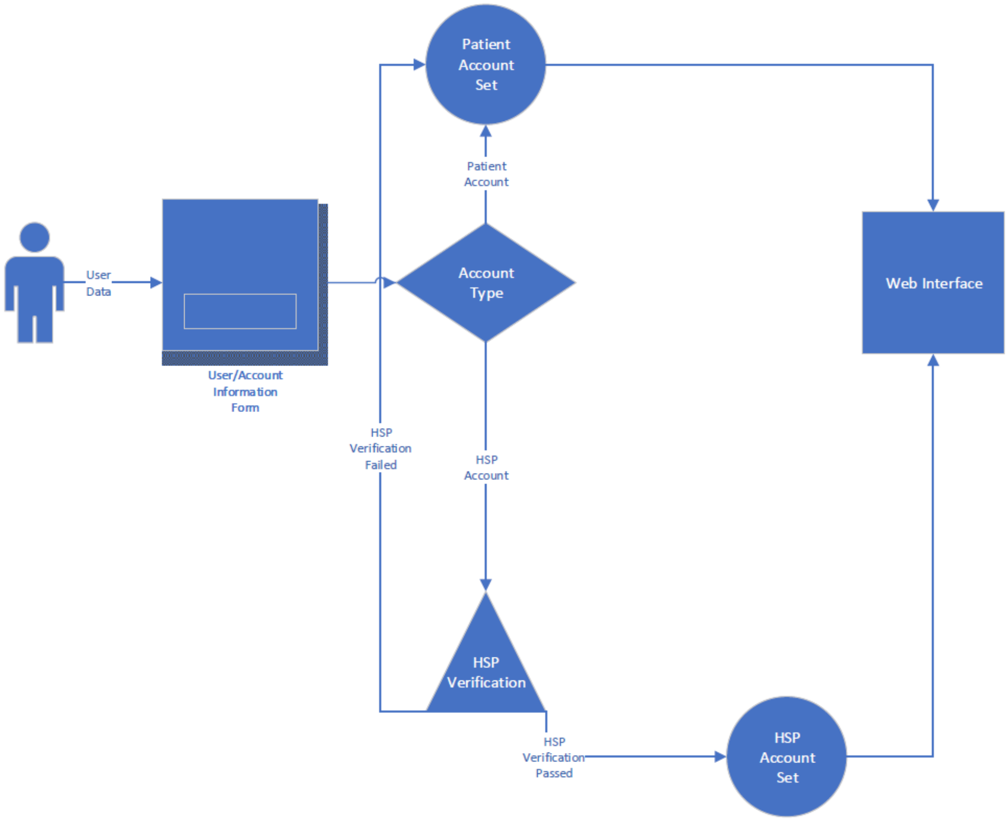


Figure 6 - Account Access Diagram

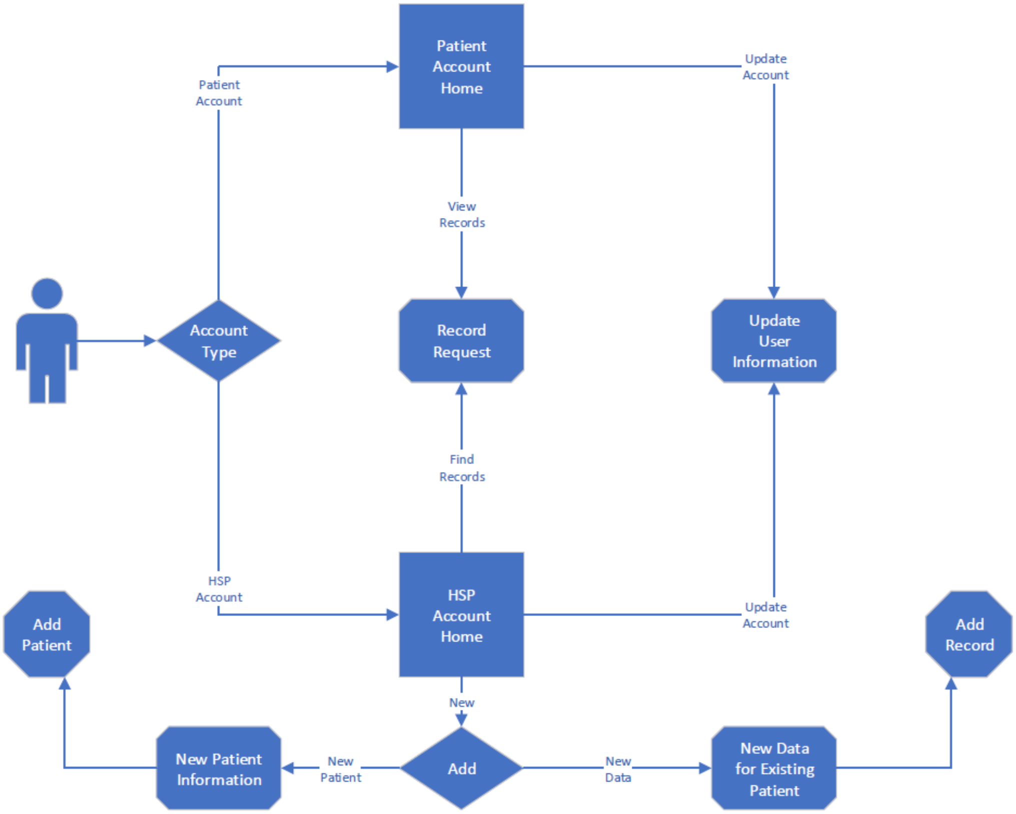


Figure 7 - Record Request Diagram

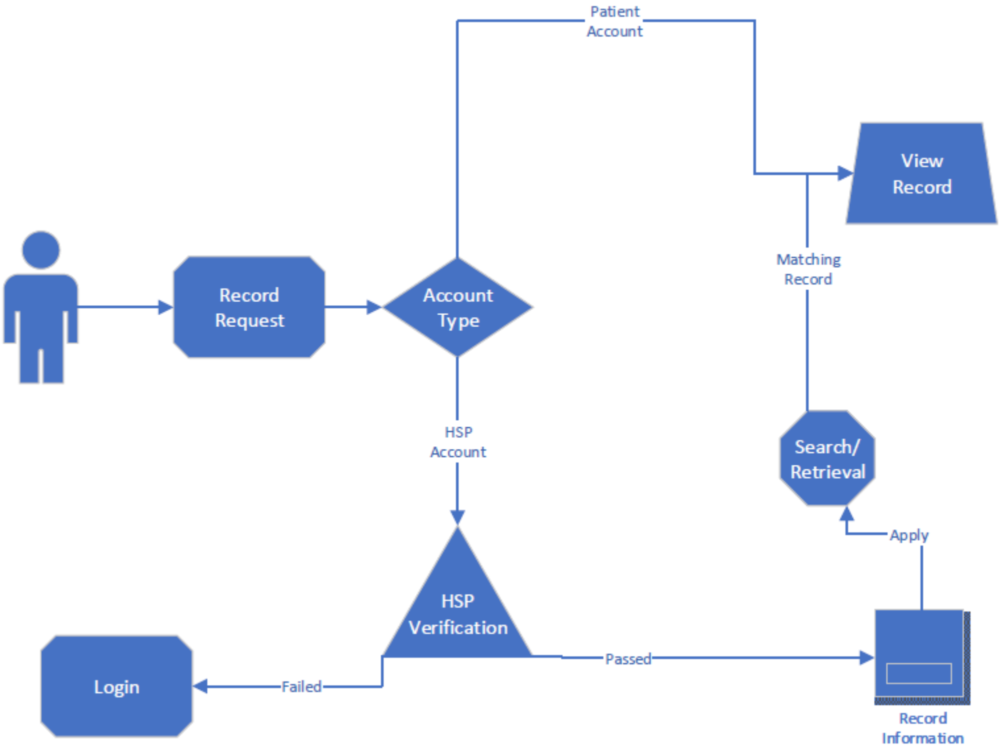


Figure 8 - Retrieval Process

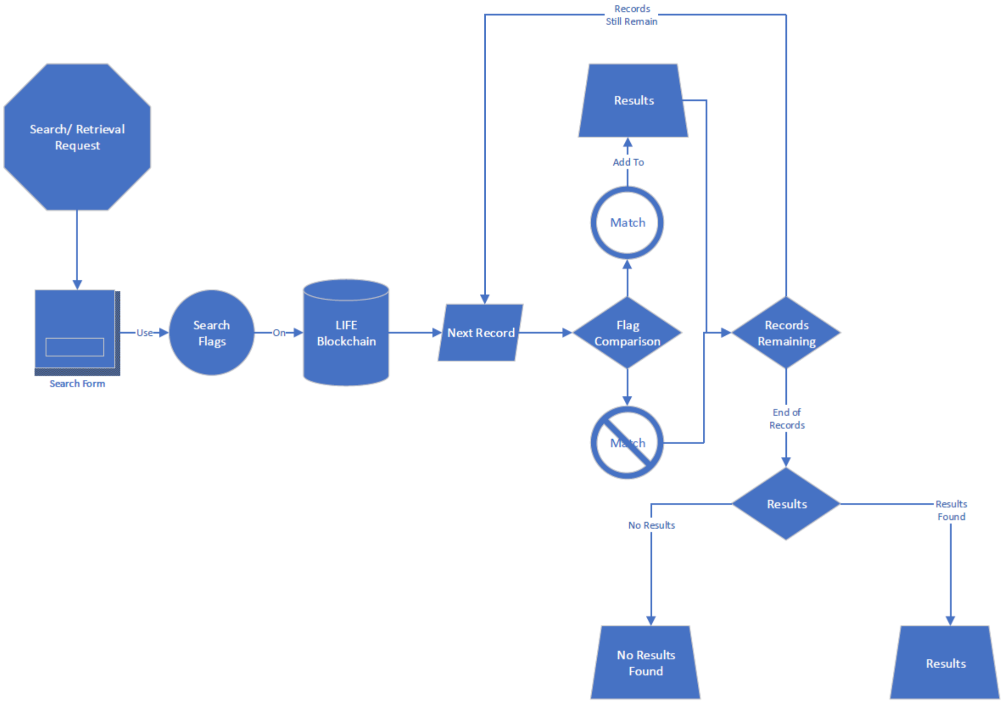


Figure 9 - Add New Patient Process

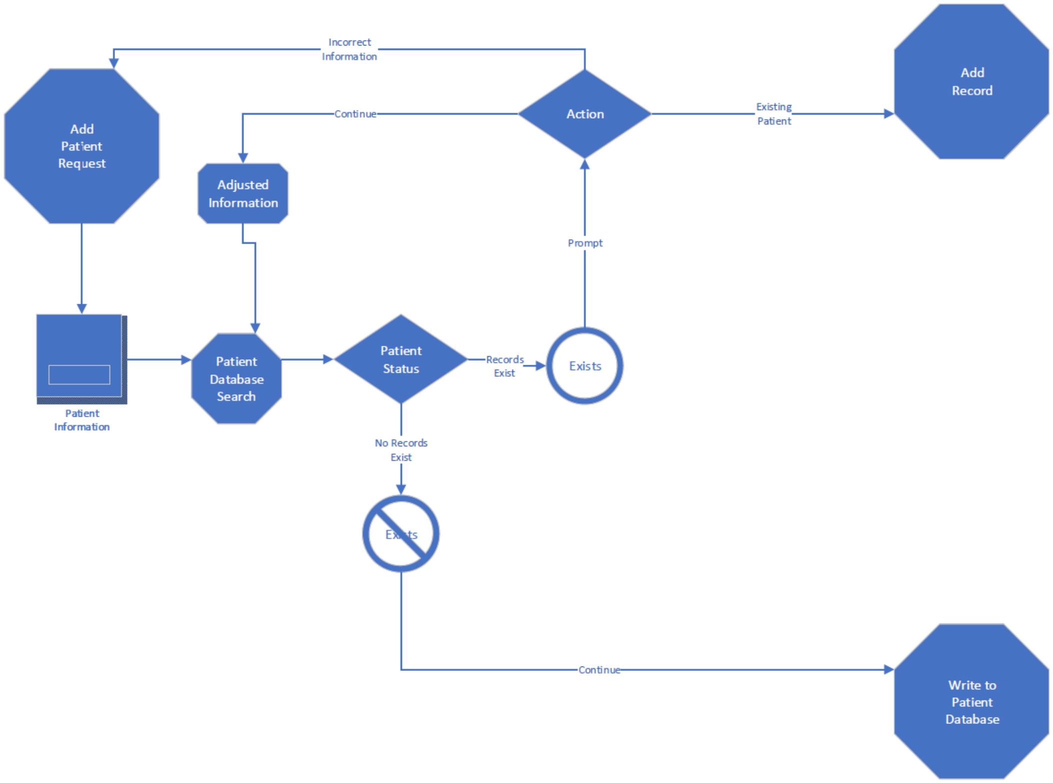


Figure 10 - Record Search Process

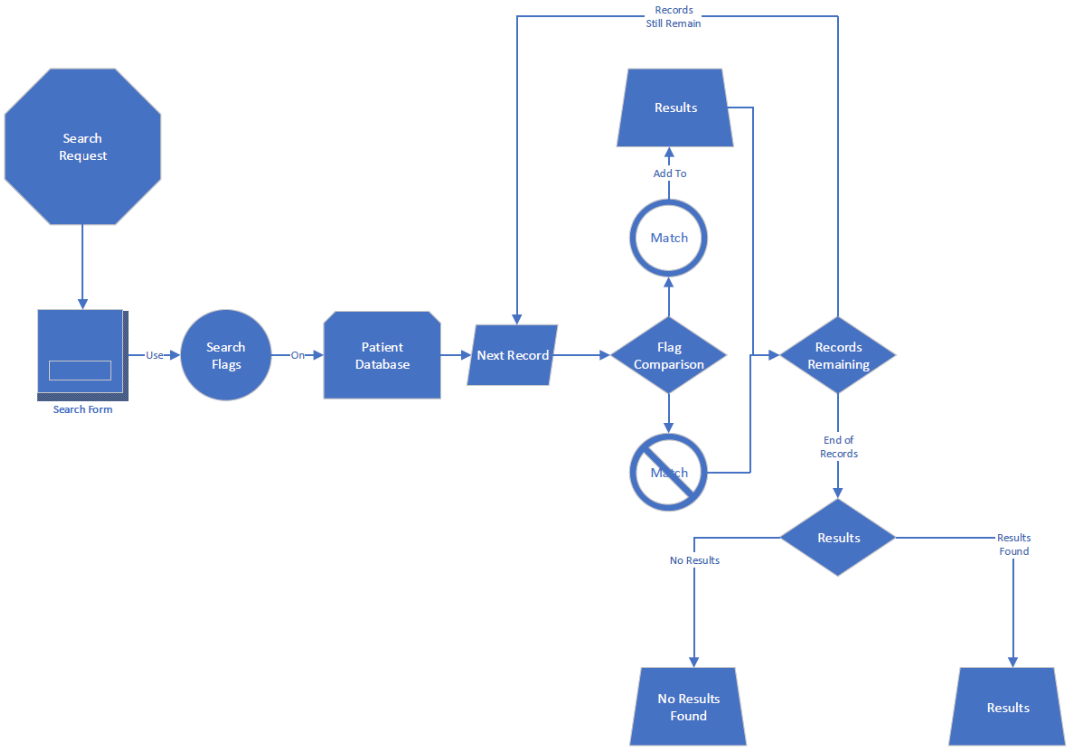


Figure 11 - Patient Creation Diagram

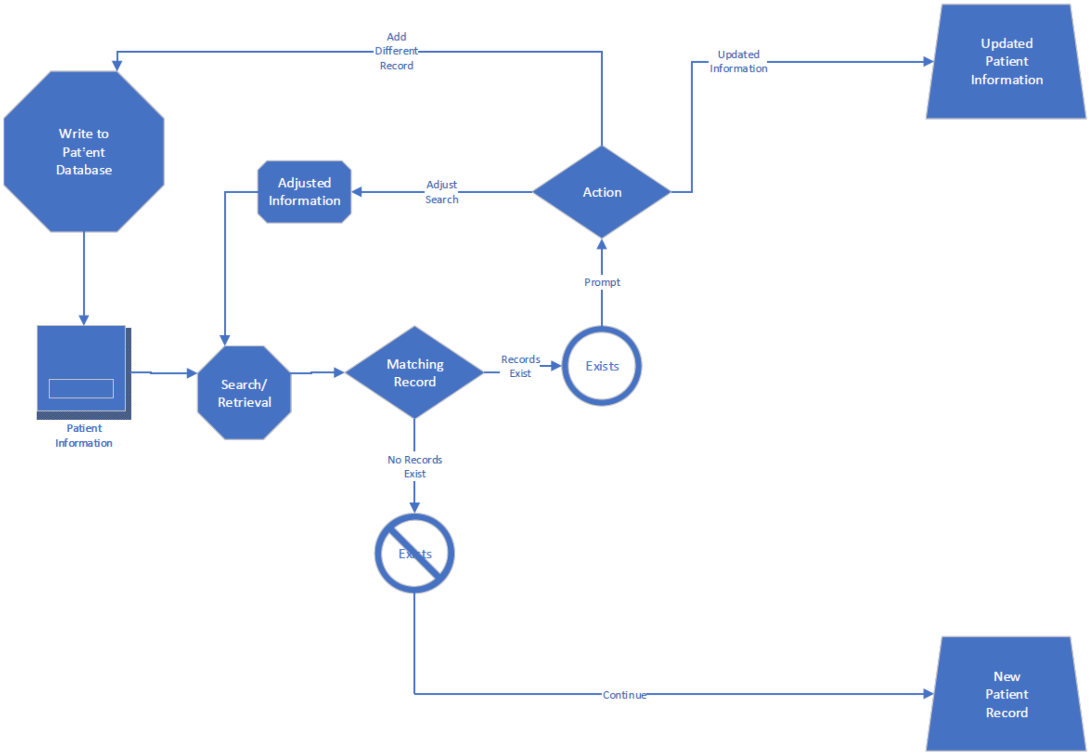


Figure 12 - New Block Entry Diagram

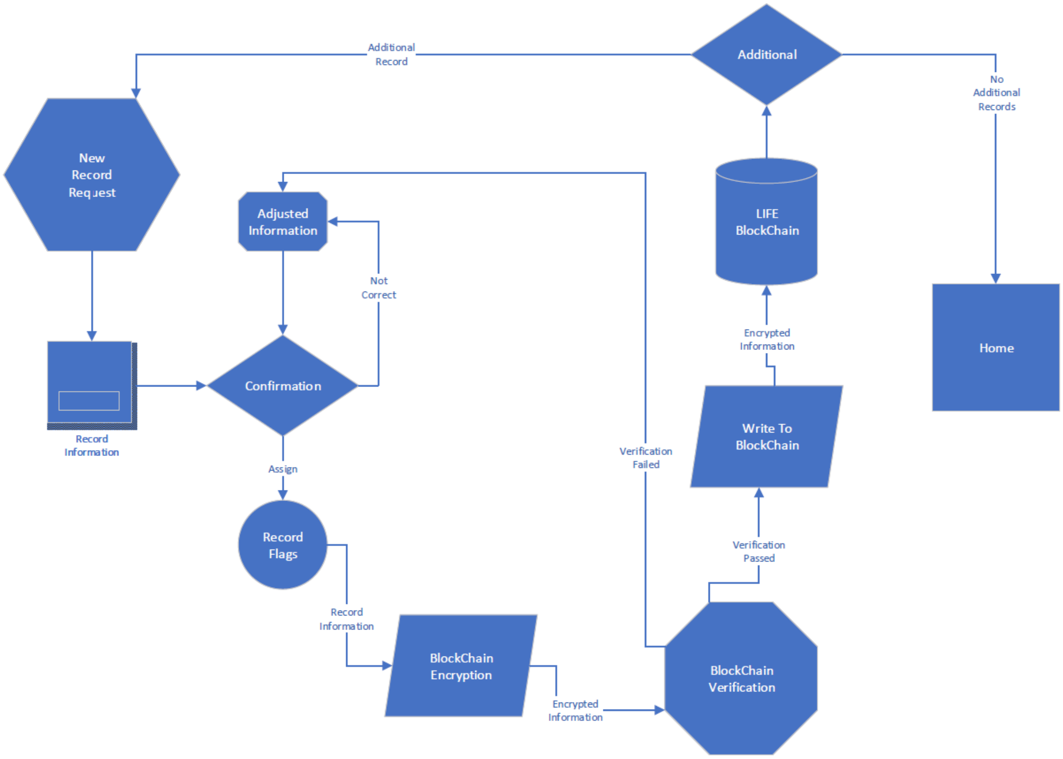
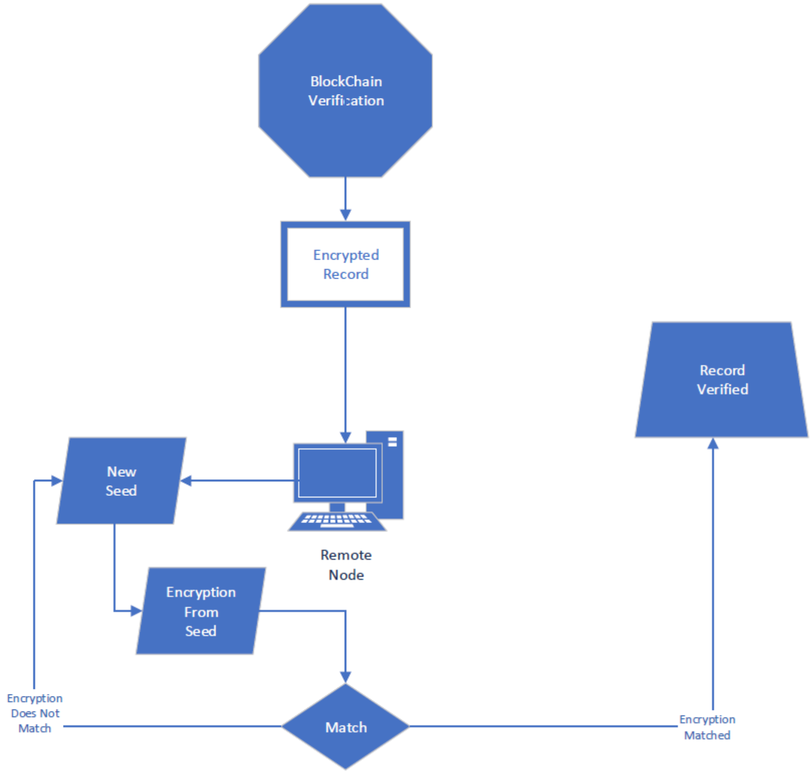


Figure 13 - Block Verification Diagram


BlockChain Verifi:ation
Encrypted Record
Record Verified
New Seed
Remote Node
Encryption From Seed
Encryption Does Not Match
Match
Encryption Matched