

CS496 Software Project Proposal: Glory Harbor Works

Crystal Ajayi & Zoe Willis

2026-02-25

1 Client Information

By sharing this client information and the rest of this document, you are stating that this client has provided this project as something they want (not something you created and asked if they wanted), and that they are interested in having you complete this project for your capstone.

1. Client name: Victor Akinde
2. Client title: Pastor of STGCI-Glory Harbor
3. Client email address: stgciglory@gmail.com
4. Client employer: STGCI-Glory Harbor
5. How you know the client: Pastor Victor oversees the church that Crystal attends.

2 Project Description

2.1 Overview

For the project, we will create a website that centralizes many important church related functions into one accessible platform. In Crystal's experience serving in church in the media department, she has seen a need for better organization, clearer communication, and streamlined scheduling. There is also a need for a more modern way to share sermons, support members through prayer requests, and engage children with age-appropriate learning materials. This website aims to solve these challenges by giving the church a structured and easy-to-use tool that strengthens community involvement and improves internal operations.

2.2 Key Features

These are the key features of the software system the client would like us to build:

- **Sermon Media Hub:** Upload and stream audio/video sermons, with the ability to search through the videos by speaker, topic, or series.
- **Prayer Request System:** Allow users to submit public or anonymous requests, interact through an "I prayed for this" feature, and enable pastors to manage and categorize incoming requests.
- **Children's Ministry Learning Center:** Provide weekly lessons, quizzes, memory verses, and progress tracking designed for younger members.
- **Pastoral Appointment Scheduling:** Allow members to book one-on-one meetings with pastors, with automated confirmations and reminders.
- **User Accounts and Roles:** Secure login system with different permissions for members, leaders, and pastoral staff.

- **Push Notifications:** Alerts for new sermons, prayer updates, scheduled meetings, or new children's lessons.

2.3 Why this Project is Interesting

When deciding what to tackle for the capstone project, we knew we wanted to create something meaningful that would have a real impact on others. This project is important because it allows us to directly support a congregation and provide a tool that improves communication, organization, and learning within the church. The challenges addressed by the website can also be used in other congregations that face similar needs in the future. The combination of practical needs, technical components, and personal motivation makes it a well rounded and exciting capstone project.

2.4 Areas of CS required

The areas of computer science the project will use includes:

- Mobile Application Development – for building a cross-platform church mobile app.
- Database Systems – for storing sermons, prayer requests, user accounts, lesson progress, and scheduling data.
- Backend/Web Services – to create APIs for authentication, sermon delivery, prayer management, and scheduling.
- Cloud Computing – for hosting media content, databases, authentication services, and storage.
- Information Security – to protect user data, prayer confidentiality, children's learning records, and secure user authentication.
- Human-Computer Interaction (HCI) – to design an accessible, intuitive interface for a diverse user base including kids, adults, and elderly members.
- Software Engineering – for requirements analysis, design, testing, documentation, and project management.

2.5 Potential Concerns and Questions

The scope of this project is a bit daunting. We will need to spend more hours working on it compared to the project for Software Engineering as there are fewer people working on it at once. We'll have to commit a considerable amount of time to it outside of the time we must spend on other assignments. In addition to that, Zoe does not come from a religious background and is uncertain if that will affect her understanding of the project.

2.6 Comparison to Draft

This project is the same one that Crystal proposed with some adjustments. It has the same client and the same basic overview. The main change of the project is the switch from an app to a website. The learning curve of trying to create an app with zero prior experience is too high for us to adapt to in the amount of time we have. Also based on the feedback from the proposal, we removed the stream audio/video sermon function and the machine learning aspect. There's a possibility they'll be too difficult to accomplish so we scrapped those ideas all together.

3 Requirements

3.1 Non-Functional Requirements

ID	NFR Title	Category	Description
NFR1	Access Control	Security	The system shall enforce role-based access control to ensure users can only access features permitted by their assigned roles.
NFR2	Session Security	Security	The system shall automatically log users out after a period of inactivity to prevent unauthorized access to accounts.
NFR3	Data Persistence	Reliability	The system shall reliably save and recover user data to prevent data loss in the event of system restarts or failures.
NFR4	Consistent Layout	Usability	The system shall maintain a consistent layout and design across all pages to improve ease of use and navigation.
NFR5	Color Scheme Consistency	Usability	The system shall follow a consistent color scheme to ensure the website is visually appealing and easy to read.
NFR6	Mobile Compatibility	Portability	The system shall be mobile-friendly and compatible with different devices and screen sizes.
NFR7	Accessibility Compliance	Accessibility	The system shall be accessible to users with disabilities by supporting assistive technologies and accessibility best practices.

Table 1: Non-Functional Requirements

3.2 Functional Requirements (User Stories)

ID	Story Title	Points	Description
M1	Create Account	2	As a Member, I want to create a profile, so that I can access the website as a member.
M2	View Account	2	As a Member, I want to view my profile, so that I can look at my information
M3	Update Account	2	As a Member, I want to update my profile, so that I can change any info if needed.
M4	Delete Account	2	As a Member, I want to delete my profile, so that I can no longer be on the website.
M5	Create Child	2	As a Member, I want to register a child under my account, so that my child can access children's lessons safely.
M6	View Child	2	As a Member, I want to view my child's learning progress, so that so that I can support their participation
M7	Update Child	2	As a Member, I want to update my child's account settings, so that so that content stays age-appropriate.
M8	Delete Child	2	As a Member, I want to deactivate a child account, so that so that accounts that are no longer needed are no longer active.
M9	Create Meeting	2	As a Member, I want to schedule meetings with one of the pastors, so that I can meet with them one on one
M10	View Meeting	2	As a Member , I want to reschedule or update meetings, so that I can adjust if plans change.
M11	Update Meeting	2	As a Member, I want to cancel meetings, so that I can adjust if plans change.
M12	Delete Meeting	2	As a Member, I want to cancel meetings, so that I can adjust if plans change.

ID	Story Title	Points	Description
M13	Create Prayer Request	2	As a/an Member , I want to create a prayer request (publicly or anonymously), so that the church can pray for me.
M14	View Prayer Request	2	As a Member, I want to view my current prayer requests, so that I can see what requests I have currently sent in and previously.
M15	Update Prayer Request	2	As a Member, I want to edit my prayer request , so that I can update information.
M16	Delete Prayer Request	2	As a Member , I want to delete my prayer request, so that the pastors/leaders never see it before it gets to them.
M17	Stream Sermons	2	As a Member, I want to stream audio and video sermons, so that I can engage with church messages remotely.
M18	Search Sermons	2	As a Member, I want to search sermons by speaker, topic, or series, so that I can easily find relevant messages.
M19	Comment on Sermon	1	As a Member, I want to comment on a sermon, so that I can share my thoughts.
M20	Update Comment	1	As a Member, I want to edit a comment, so that I can change what I commented.
M21	Delete Comment	1	As a Member, I want to delete a comment on a sermon, so that I can remove my comment if I want.
M22	Like a Sermon	1	As a Member, I want to like a sermon, so that I can share my thoughts.
M23	Remove Like	1	As a Member, I want to unlike a sermon, so that I can share my thoughts.
M24	Log In	1	As a Member, I want to log into my account, so that I can access specialized features.
M25	Log Out	1	As a Member, I want to log out of my account, so that I keep my information secure.
M26	Reset Password	2	As a Member, I want to reset my password, so that I can regain access if I forget it.
M27	Send Pastoral Appointment Reminders	1	As a Member, I want to receive reminders for pastoral appointments, so that I do not forget.
M28	Push Notifications	1	As a Member, I want to receive push notifications, so that I can be alerted for new sermons, prayer updates, scheduled meetings, or new children's lessons.
P1	Availability Input	2	As a Pastor, I want to enter my availability, so that I can be scheduled efficiently.
P2	Approve/Decline Meetings	2	As a Pastor, I want to approve or decline meeting requests, so that my schedule is manageable.
P3	View Schedule	2	As a Pastor, I want to view my scheduled meetings, so that I can meet with members on time.
P4	Cancel Meetings	2	As a Pastor, I want to cancel scheduled meetings, so that changes are reflected.
P5	View Prayer Requests	2	As a Pastor, I want to view all incoming prayer requests, so that I can complete them.
P6	Mark Requests as Done	2	As a Pastor, I want to mark prayer requests as complete, so that I can let the member know I have completed their request.
C1	Weekly Lessons	2	As a Child, I want to view weekly lessons and memory verses, so that I can learn about faith in a fun and engaging way.
C2	Complete Quizzes	3	As a Child, I want to complete quizzes after lessons, so that I can practice and remember what I learned.

ID	Story Title	Points	Description
C3	Learning Progress	3	As a Child, I want to see my learning progress, so that I feel motivated to continue participating.
L1	Upload Audio Sermon	3	As a Leader, I want to upload audio sermons, so that I can engage with church messages remotely.
L2	Upload Video Sermon	3	As a Leader, I want to upload video sermons, so that I can engage with church messages remotely.
L3	Edit Sermon	3	As a Leader, I want to edit uploaded sermons, so that I can keep the sermons up to date.
L4	Delete Sermon	2	As a Leader, I want to delete uploaded sermons, so that the content stays current.
L5	Deactivate Accounts	1	As a Leader, I want to deactivate unused accounts, so that the current member accounts stay updated.
L6	Assign Roles	1	As a Leader, I want to assign roles to users, so that permissions are managed correctly.
L7	Send Announcement	1	As a Leader, I want to send announcements or notifications, so that members stay informed.
G1	View Page	1	As a Guest, I want to view the landing page, so that I understand the church's mission.
G2	Stream Sermon	1	As a Guest, I want to stream audio and video sermons, so that I can engage with church messages remotely.
G3	View Info	1	As a Guest, I want to view information about the church, so that I can learn about the church.
G4	Social Media	1	As a Guest, I want to view the church's different social media accounts, so that I can stay connected with them on different platforms.
G5	Contact Church	2	As a Guest, I want to contact the church, so that I can ask questions easily.

Table 2: Functional requirements as User Stories

4 System Design

4.1 Architecture

We will use the MVC structure to define how different components interact in our system. We like this model for our project because it is good at defining roles when users interact with data. The Model is what manages the data. The controller is the messenger between the model and the view and it handles the incoming requests using logic. The view is the interface that interacts with the user. The main modules would be User Management, Sermons, Prayer Requests, Scheduling, Children's Ministry, and Notifications.

4.2 Diagrams

The View points to the Model because the View needs to read data from the Model in order to display up-to-date information to the user. The Controller points to the Model because it is responsible for updating and managing that data and logic in response to user actions. The Controller points to the View because it controls which views are updated after processing user input, ensuring the user sees the current state of the system.

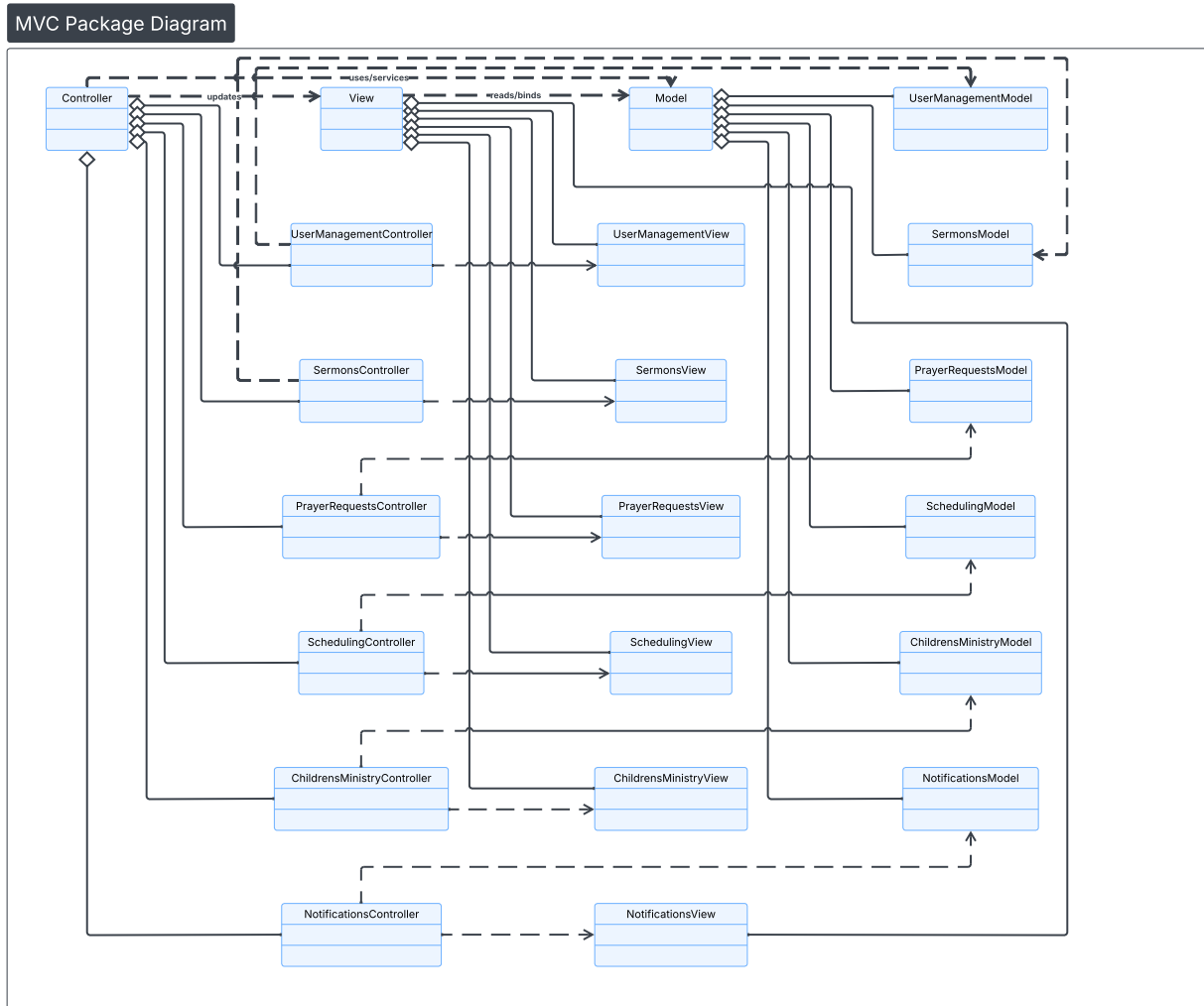


Figure 1: MVC Package Diagram for the Church Website Architecture

4.3 Technology

Table 3: Technology Stack Overview

Technology	Purpose and Justification
JavaScript	Main programming language. Used to implement controllers and Data Access Objects (DAOs) in a consistent and efficient manner. Supports clean code organization and aligns with the MVC architecture.
HTML	Frontend markup language. Chosen due to prior experience, enabling efficient development. Works seamlessly with CSS to maintain consistent page structure.
CSS	Used for styling and layout customization. Ensures a consistent look and feel across all pages.

Technology	Purpose and Justification
React	Frontend library used to improve code re-usability, modularity, and web-page performance through component-based architecture.
Node.js	Backend runtime environment that executes JavaScript server-side.
Express.js	Backend framework used with Node.js to handle routing and manage web requests efficiently.
MongoDB	NoSQL database used to store application data in a flexible, document-based structure.
Mongoose	Object Data Modeling (ODM) library used to define schemas and enforce structure while maintaining MongoDB flexibility.
JSON	Data formatting standard used for structuring and transferring data between frontend and backend.
Playwright	Primary testing framework used to test authentication, sermon streaming, scheduling, and role-based access functionality.

4.4 Coding Standards

We will be following coding standards to ensure we maximize productivity between the 2 of us. We will follow our iteration planning that we decided on in the beginning of our work. We will follow the standard that we will stick to our iteration planning to our best ability, and if a change is needed to be made, we will consult our partner and correctly reflect the change in our documentation. We will release and push our code frequently, meaning that anytime a significant amount of progress in a user story is made or a user story is completed we will commit and push. We will require that every feature be accompanied by unit tests, and no code will be released unless all unit tests pass with at least 60% coverage. This ensures correctness, reliability, and prevents regressions throughout development. We will use camelCase for all javascript code and database collections will use lowercase singular names. To reduce merge conflicts, we will clearly divide tasks and communicate frequently to avoid working on the same files at the same time.

4.5 Data

We will be using a NoSQL database, as it is best suited for web applications and projects that evolve quickly due to its flexible schema structure. We will utilize MongoDB to store our data and take advantage of its schema flexibility as the project grows and requirements change. Mongoose will be used as an Object Data Modeling (ODM) library to allow our backend code to communicate efficiently with the database while still enforcing structure where necessary.

Collection	Key Attributes / Description
Members	ID, Email, First Name, Last Name, Role, Status
Leaders	Extends Member. Includes all Member fields plus: Managed Members, Scheduling Availability
Pastors	Extends Member. Includes all Member fields plus: Managed Members, Scheduling Availability
Children	Child-specific information (linked to Member accounts as needed).
Sermons	ID, Title, Speaker, Topic, Type, URL
Prayer Requests	ID, CreatedBy (nullable for anonymous submission), Text, Status
Appointments	ID, MemberID, PastorID, Time, Location, Status
Notifications	ID, UserID, Type, Contact Method, Message, TimeSent

Table 4: Database Collections and Key Attributes

4.6 UI Mocks



Figure 2: System screenshots showing key features of the application. The landing page, sign in page, and sermon media hub.

5 Iterations

5.1 Iteration Planning

Table 5 shows the iteration planning.

Iteration	Dates	Stories	Points
1	01/27 – 02/10	G1 View Page, G2 Stream Sermon, G3 View Info, G4 Social Media, G5 Contact Church, M1 Create Account, M2 View Account, M3 Update Account, M4 Delete Account, M24 Log In, M25 Log Out, NFR1 Access Control, NFR4 Consistent Layout	16
2	02/11 – 03/24	L1 Upload Audio Sermon, L2 Upload Video Sermon, M17 Stream Sermons, M18 Search Sermons, M22 Like Sermon, M23 Remove Like, NFR5 Color Scheme	12
3	03/25 – 03/17	M13 Create Prayer Request, M14 View Prayer Request, M15 Update Prayer Request, M16 Delete Prayer Request, P5 View Prayer Requests, P6 Mark Requests as Done, M27 Send Pastoral Appointment Reminders, NFR2 Session Security	13
4	03/18 – 03/31	M9 Create Meeting, M10 View Meeting, M11 Update Meeting, M12 Delete Meeting, P1 Availability Input, P2 Approve/Decline Meetings, P3 View Schedule, P4 Cancel Meetings, NFR3 Data Persistence	16
5	04/01 – 04/14	M5 Create Child, M6 View Child, M7 Update Child, M8 Delete Child, C1 Weekly Lessons, C2 Complete Quizzes, C3 Learning Progress, L5 Deactivate Accounts, L6 Assign Roles, L7 Send Announcement, M19 Comment on Sermon, M20 Update Comment, M21 Delete Comment, M28 Push Notifications, NFR6 Mobile Friendly, NFR7 Accessibility	17
Total:			84

Table 5: Iteration Planning for Incremental Deliveries

5.2 Iteration/Sprint 1

5.2.1 Planning

For Iteration 1, our team planned a total of 16 story points. The objective of this iteration was ensuring the implementation of access to the core content, authentication, and user accounts, which enable the user to log in and out of the system completely.

In the case of guests, we included features like View Page, Stream Sermon, View Info, Social Media, Contact Church to ensure maximum user engagement even for users who have not registered on the site. These features clearly define the main purpose of the website to users.

At the same time, we concentrated on the essential members’ functionality which includes implementing the “Create Account,” “Log In,” “Log Out,” and full lifecycle management (“view,” “update,” “delete”) functionalities of a member’s account, which are precursors to the advanced members’ functionalities we intend to design in the future.

We also plan to add essential non-functional features like Access Control and Consistent Layout to ensure correct access control for different users’ roles, as well as providing an overall user-friendly interface for the entire application. With the implementation of these stories, this iteration delivers a stable, secure, and user-friendly baseline that future features can reliably build upon.

5.2.2 Work Done

Zoe completed stories M1-M3 for a total of 6 story points. M4 can only be done by admin at the time so that story is only partially completed at the moment, for one more story point.

Crystal completed G1-G5 for a total of 6 points and M24-M25 for 2 points. We both had contributions to access control and Crystal assured our landing page has consistent layout.

5.2.3 Testing Coverage

At the moment, our coverage is not good enough. Our coding standard targets 60% coverage before release. In Iteration 1, we focused on establishing the project structure and verifying core processes like auth middleware and the user model. Overall coverage is currently below the 60% target because several modules were framed but not fully implemented/tested yet. In Iteration 2, we will prioritize adding integration tests for all auth and user endpoints and expand frontend component tests so that overall coverage increases.

```
PS C:\Users\z4rai\OneDrive - Loyola University Maryland\CS 496\glory-harbor-works\server> npm run test:coverage
> server@1.0.0 test:coverage
> node --experimental-vmodules node_modules/jest/bin/jest.js --coverage

(node:24468) ExperimentalWarning: VM Modules is an experimental feature and might change at any time
(Use `node --trace-warnings ...` to show where the warning was created)
PASS tests/middleware/auth.test.js
  Auth Middleware
    ✓ should return 401 if no authorization header provided (8 ms)
    ✓ should be a function

(node:11260) ExperimentalWarning: VM Modules is an experimental feature and might change at any time
(Use `node --trace-warnings ...` to show where the warning was created)
PASS tests/models/User.test.js
  User Model
    User Schema Validation
      ✓ should create user with valid data (13 ms)
      ✓ should set default role to member (1 ms)
      ✓ should set default status to active
      ✓ should convert email to lowercase (1 ms)
      ✓ should have comparePassword method (1 ms)
      ✓ should require email, firstName, lastName, and password (4 ms)

(node:9776) ExperimentalWarning: VM Modules is an experimental feature and might change at any time
(Use `node --trace-warnings ...` to show where the warning was created)
PASS tests/controllers/authController.test.js
  Auth Controller
    register
      ✓ should be a function (6 ms)
    login
      ✓ should be a function (1 ms)
    logout
      ✓ should return logout success message
      ✓ should be a function (1 ms)

-----|-----|-----|-----|-----|-----
File    | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----|-----|-----|-----|-----|-----
...iles | 8.86    | 1.36     | 11.76    | 8.91    |
...ers  | 4.22    | 0        | 7.14     | 4.22    |
...js   | 14.28   | 0        | 20       | 14.28   | ...66-101,107-114
...js   | 0       | 0        | 0        | 0       | 7-354
...are  | 55.55   | 25       | 100      | 55.55   |
...js   | 55.55   | 25       | 100      | 55.55   | 11-15
models  | 42.85   | 0        | 0        | 50      |
...js   | 42.85   | 0        | 0        | 50      | 29-30,35
-----|-----|-----|-----|-----|-----
Test Suites: 3 passed, 3 total
Tests:       12 passed, 12 total
Snapshots:   0 total
Time:        2.139 s
Ran all test suites.
```

Figure 3: Iteration 1 test coverage report.

5.2.4 Retroespective & Reflection

Zoe - When setting up the database in MongoDB I didn't know I had to set access to everywhere, so at first Crystal couldn't access it. It was a minor setback, but something that could have easily been avoided if I had read the directions or watched a video on how to set it up. Currently I have to run the server and client side with different commands and in different terminals so that is something we need to improve in the next iteration.

Crystal - As I work on the website, one of my main focuses is the aesthetic. I want the site to be visually appealing or at least look properly organized. I spent most of my time on the UI. I also worked on the registration page but ran into trouble when I tried to test it. Once I got access from Zoe everything worked

properly. I am enjoying implementing different UI features but I need to make sure I spend time on the actual user stories and not just the looks.

5.3 Iteration/Sprint 2

5.3.1 Planning

We plan to complete L1 Upload Audio Sermon, L2 Upload Video Sermon, M17 Stream Sermons, M18 Search Sermons, M22 Like Sermon, M23 Remove Like, and NFR5 Color Scheme this iteration for a total of 12 story points. This iteration is mainly focused on complete the sermon feature of our main modules in our MVC structure. We may implement more stories if time permits as this is our lowest point value iteration that we have planned, and our last iteration has the most points so we should do more now when we can.

5.3.2 Work Done

Zoe completed M17 to stream sermons, M18 to search sermons, M22 Like Sermon, and M23 Remove Like for a total 6 story points. M17 was to create a sermon hub and add navigation to it. Now, the sermons act as more of a chat box than just youtube videos embedded on the main page.

Crystal completed L1 Upload Audio Sermon, L2 Upload Video Sermon, and G3 View Info from the last iteration as well as some fixes/design changes for a total of 7 points. Authorized users can upload and edit sermons and logged in users are able to listen to the sermons and comment on them. Users are also able to upload a pfp that is visible when they comment.

5.3.3 Testing Coverage

Our coverage is much better than the last iteration. Although, we can still improve much more especially in branch coverage. We need to add more targeted method branch tests to increase User.js coverage.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	32.55	20.39	39.21	32.87	
controllers	31.27	21.42	41.46	31.55	
authController.js	14.28	0	20	14.28	7,14-62,68-105,111-118
contactController.js	79.31	86.66	100	79.31	34-35,58-59,91-92
prayerController.js	0	0	0	0	4-87
sermonController.js	19.7	10.25	29.41	20.1	...9-240,269-337,355-377,386-435,440-483,499-535,555-578
userController.js	50.69	41.33	66.66	50.69	...2,257,263,275-276,289-330,355-367,393-399,432,444-445
middleware	38.29	9.09	37.5	38.29	
auth.js	30	8.33	50	30	12-16,21-31,41-57
upload.js	52.94	10	25	52.94	11,17-22,27-31
models	53.33	0	0	57.14	
Appointment.js	0	100	100	0	3
Contact.js	100	100	100	100	
Notifications.js	0	100	100	0	3
Prayer.js	0	100	100	0	3
Sermon.js	100	100	100	100	
Prayer.js	0 100 100 0 3				Prayer.js
Prayer.js	0	100	100	0	3
Sermon.js	100	100	100	100	
User.js	42.85	0	0	50	30-31,36
Test Suites: 12 passed, 12 total					
Tests: 67 passed, 67 total					
Snapshots: 0 total					
Time: 3.419 s					
Can all test suites.					

Figure 4: Iteration 2 test coverage report.

5.3.4 Retrospective & Reflection

Zoe - The biggest challenge for us for this iteration was testing. We had to catch up on implementing all testing from the last iteration, and it piled up and ended up being the most time consuming task this iteration. We realized that we need to be writing our tests as we implement different modules of our webpage so that we can test after we build a certain feature and not wait for the end to do all testing. Also we could've added more points to this iteration rather than having more points in our last iteration.

Crystal - This iteration didn't include as many points as the previous one. Although we completed everything that was planned, we should have worked on upcoming iterations to lessen our future workload. Going

forward, I will begin stories earlier so I have enough time to polish the UI, make sure everything works properly and, when possible, work on future iterations.

5.4 Iteration/Sprint 3

5.4.1 Planning

[Which stories did you plan for this iteration/sprint. Add the total points for this plan. You can also explain the reason behind your planning, and what major feature(s) your team is focusing on delivering by completing these stories. You may use a table for a summary display of the planning, but elaborate in text more detail in your focus and feature plan.]

5.4.2 Work Done

[Which stories did you complete in this iteration/sprint. Which ones did you partially complete? Who worked on which story? You may elaborate in paragraph(s) to add more detail about the work done.]

5.4.3 Testing Coverage

[Testing is very important. Show your coverage here. Is this coverage good enough? Explain why you think so. Is it not good enough? Explain a plan to increase the coverage. You may also elaborate on why some artifacts do not undergo much testing. If the testing changed from the last iteration, explain the reasons.]

5.4.4 Retroerspective & Reflection

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]

5.5 Iteration/Sprint 4

[CS496 has 5 sprints. CS482 only has only 3 sprints (remove Iterations 4 and 5 from this doc if you are writing a doc for 482)]

5.5.1 Planning

[Which stories did you plan for this iteration/sprint. Add the total points for this plan. You can also explain the reason behind your planning, and what major feature(s) your team is focusing on delivering by completing these stories. You may use a table for a summary display of the planning, but elaborate in text more detail in your focus and feature plan.]

5.5.2 Work Done

[Which stories did you complete in this iteration/sprint. Which ones did you partially complete? Who worked on which story? You may elaborate in paragraph(s) to add more detail about the work done.]

5.5.3 Testing Coverage

[Testing is very important. Show your coverage here. Is this coverage good enough? Explain why you think so. Is it not good enough? Explain a plan to increase the coverage. You may also elaborate on why some artifacts do not undergo much testing. If the testing changed from the last iteration, explain the reasons.]

5.5.4 Retrospective & Reflection

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]

5.6 Iteration/Sprint 5

5.6.1 Planning

[Which stories did you plan for this iteration/sprint. Add the total points for this plan. You can also explain the reason behind your planning, and what major feature(s) your team is focusing on delivering by completing these stories. You may use a table for a summary display of the planning, but elaborate in text more detail in your focus and feature plan.]

5.6.2 Work Done

[Which stories did you complete in this iteration/sprint. Which ones did you partially complete? Who worked on which story? You may elaborate in paragraph(s) to add more detail about the work done.]

5.6.3 Testing Coverage

[Testing is very important. Show your coverage here. Is this coverage good enough? Explain why you think so. Is it not good enough? Explain a plan to increase the coverage. You may also elaborate on why some artifacts do not undergo much testing. If the testing changed from the last iteration, explain the reasons.]

5.6.4 Retrospective & Reflection

[What were the pitfalls, challenges, and issues you had in this iteration? How can you address them to improve the process in the next iteration? Did anything not go according to plan? Why so and how to avoid the same mistake? Write a personal reflection on what you learned in this iteration (even if a small technical thing like Database storage).]

6 Final Remarks

6.1 Overall Progress

[Have you completed everything? If so, present evidence on how you brought value to your client, and the overall client satisfaction. Otherwise, estimate how much progress you done and how long it would take to finish this project. Be concrete about your progress, you know how many story points your software is, how many points you completed (this shows your progress). You also how many points your team delivers at each iteration, therefore you can estimate how many more iterations it would take to finish the leftover points (show the math).]

6.2 Project Reflection

[Your personal reflection on the project. What lessons did you learned. What would you have done differently? How can you do better work in future projects? You may write this as a team or per person (or both — if all your iterations were team reflections, then it would be better to write individual reflections here)]

Appendix

[Appendix section if needed]