Week 1 Class Notes: Building the Ransem Foundation Website

AI-Powered Web Development Course

Session 1: Requirements Engineering & Client Understanding

Learning Objectives

By the end of this session, you will be able to:

- Explain the Software Development Life Cycle (SDLC) and its key phases
- Identify and analyze different types of stakeholders in a web project
- Distinguish between functional and non-functional requirements
- Create a comprehensive Software Requirements Specification (SRS) document
- Use AI tools to analyze client briefs and generate user stories effectively

Detailed Content Outline

1. Software Development Life Cycle (SDLC) Overview (20 minutes)

What is SDLC? The Software Development Life Cycle is a structured approach to developing software that ensures quality, efficiency, and client satisfaction. Think of it as a roadmap that guides us from the initial idea to a fully functional website.

Key SDLC Phases:

1. **Planning & Requirements Gathering** ← *We are here!*
2. **Design & Architecture**
3. **Development & Implementation**
4. **Testing & Quality Assurance**
5. **Deployment**
6. **Maintenance & Support**

Why SDLC Matters for Web Development:

- Prevents scope creep and miscommunication
- Ensures all stakeholder needs are addressed
- Creates a clear timeline and budget framework

- Reduces development costs and time-to-market

2. Stakeholder Analysis (25 minutes)

What are Stakeholders? Stakeholders are anyone who has an interest in or will be affected by your web project. Understanding them is crucial for building the right solution.

Types of Stakeholders:

- **Primary:** Direct users of the system (donors, volunteers, beneficiaries)
- **Secondary:** Indirect users (staff, board members, partner organizations)
- **Key:** Decision makers and influencers (foundation director, IT manager)

Ransem Foundation Stakeholder Analysis:

- **Donors:** Want transparency, impact stories, easy donation process
- **Foundation Staff:** Need content management, donor tracking, event coordination
- **Beneficiaries:** Seek program information, application processes, community connection
- **Volunteers:** Look for opportunities, scheduling, resource access
- **Board Members:** Require governance tools, financial reporting, strategic oversight

3. Functional vs Non-Functional Requirements (25 minutes)

Functional Requirements - *What the system should do*

- User registration and login
- Online donation processing
- Event management and RSVP
- Content management system
- Volunteer opportunity posting
- Impact story publishing

Non-Functional Requirements - *How the system should perform*

- **Performance:** Page load time under 3 seconds
- **Security:** PCI DSS compliance for payments

- **Usability:** Mobile-responsive design
- **Accessibility:** WCAG 2.1 AA compliance
- **Scalability:** Handle 10,000+ concurrent users
- **Reliability:** 99.9% uptime

Practical Activities

Activity 1: Stakeholder Mapping Workshop (30 minutes)

Working in teams of 3-4, students will:

1. Review the provided Ransem Foundation documentation
2. Create a comprehensive stakeholder map using sticky notes or digital tools
3. Prioritize stakeholders by influence and interest level
4. Present findings to the class for peer feedback

Activity 2: Requirements Elicitation (45 minutes)

Students will practice requirements gathering by:

1. Role-playing stakeholder interviews (one student as foundation director, others as developers)
2. Documenting functional requirements using user story format
3. Identifying and categorizing non-functional requirements
4. Creating a requirements traceability matrix

How to Use AI (Claude/ChatGPT) in This Session

AI Prompt Templates for Requirements Analysis

1. Client Brief Analysis:

"I'm working on a nonprofit foundation website project. Here's the client brief: [paste brief]. Help me:

1. Identify all potential stakeholders
2. Extract key functional requirements
3. Suggest non-functional requirements I might have missed
4. Highlight any ambiguities that need clarification"

2. User Story Generation:

"Based on this stakeholder analysis for a foundation website: [describe stakeholders], generate 15-20 user stories in the format: 'As a [user type], I want [functionality] so that [benefit]'. Focus on donation, volunteering, and program management features."

3. Requirements Validation:

"Review this list of requirements for a foundation website: [paste requirements]. Are there any:

- Conflicting requirements?
- Missing critical features for nonprofits?
- Requirements that seem too vague or too specific?
- Compliance or accessibility considerations I've overlooked?"

AI Best Practices for This Session

- Always provide context about the nonprofit sector when prompting
- Ask AI to explain its reasoning for suggestions
- Use AI to generate multiple alternatives, then apply your judgment
- Validate AI suggestions against real-world nonprofit website examples

Recommended Tools/Resources

Essential Reading

- IEEE Guide to Software Requirements Specifications
- Nonprofit Website Requirements Checklist

Video Resources

- "Requirements Engineering Fundamentals" - Software Engineering Institute (15 min)
- "Stakeholder Analysis for IT Projects" - ProjectManager.com (12 min)

Tools for Requirements Management

- **Trello/Notion:** For organizing and tracking requirements
- **Miro/Figma:** For stakeholder mapping and visual documentation
- **Google Forms:** For stakeholder surveys and feedback collection

Sample Documents

- SRS Template for Nonprofit Websites
- Stakeholder Analysis Template
- Requirements Traceability Matrix Template

Mini Quiz - Session 1 Self-Check

1. **Multiple Choice:** Which SDLC phase are we focusing on in this session?
    - a) Design & Architecture
    - b) Planning & Requirements Gathering ✓
    - c) Development & Implementation
    - d) Testing & Quality Assurance
2. **Short Answer:** Name three types of stakeholders for the Ransem Foundation website and explain why each is important.
3. **Scenario:** A foundation director says "I want a modern website that's user-friendly." Identify what's missing from this requirement and how you would clarify it.
4. **True/False:** Non-functional requirements describe what the system should do, while functional requirements describe how it should perform. (False - it's the opposite)

Session 2: Design Thinking & Wireframing

Learning Objectives

By the end of this session, students will be able to:

- Apply design thinking methodology to web development projects
- Create detailed user personas based on stakeholder research
- Map user journeys for different foundation website workflows
- Design effective wireframes using established UX principles
- Leverage AI tools to generate and validate design concepts

Detailed Content Outline

1. Introduction to Design Thinking (20 minutes)

What is Design Thinking? Design thinking is a human-centered approach to innovation that focuses on understanding users, challenging assumptions, and creating solutions that truly meet user needs.

The 5 Stages of Design Thinking:

1. **Empathize:** Understand your users deeply
2. **Define:** Frame the problem clearly
3. **Ideate:** Generate creative solutions
4. **Prototype:** Build testable representations
5. **Test:** Gather feedback and iterate

Design Thinking in Web Development:

- Ensures websites solve real user problems
- Reduces development rework and user frustration
- Creates more engaging and effective user experiences
- Aligns technical solutions with business goals

2. Creating User Personas (30 minutes)

What are User Personas? User personas are fictional characters that represent your real users. They help you make design decisions by keeping user needs at the center of your process.

Elements of Effective Personas:

- **Demographics:** Age, location, occupation, income level
- **Goals:** What they want to achieve on your website
- **Pain Points:** Current frustrations and challenges
- **Behaviors:** How they typically interact with websites
- **Technology Comfort:** Device preferences and technical skills
- **Motivations:** What drives their engagement with your cause

Ransem Foundation Persona Examples:

Persona 1: Sarah the Recurring Donor

- Age: 45, Marketing Director, $75K income

- Goals: Track donation impact, set up recurring gifts, share cause with friends
- Pain Points: Busy schedule, wants transparency in fund usage
- Behaviors: Mobile-first, social media active, prefers quick transactions
- Quote: "I want to know my money is making a real difference."

Persona 2: Michael the Volunteer Coordinator

- Age: 28, Recent college graduate, part-time work
- Goals: Find meaningful volunteer opportunities, connect with like-minded people
- Pain Points: Limited transportation, variable schedule
- Behaviors: Research-heavy, reads reviews, prefers detailed information
- Quote: "I want to use my skills to help, but need flexibility."

3. User Journey Mapping (25 minutes)

What is User Journey Mapping? User journey mapping visualizes the complete experience a user has while interacting with your website, from initial awareness to goal completion.

Key Components of User Journeys:

- **Touchpoints:** Where users interact with your site
- **Actions:** What users do at each step
- **Emotions:** How users feel throughout the process
- **Pain Points:** Where users get frustrated or confused
- **Opportunities:** Where you can improve the experience

Example Journey: First-Time Donor

1. **Awareness:** Sees social media post about foundation
2. **Interest:** Visits website homepage
3. **Research:** Reads about programs and impact
4. **Decision:** Decides to make donation
5. **Action:** Completes donation form
6. **Confirmation:** Receives thank you and receipt

7. **Follow-up:** Gets impact updates via email

4. Wireframing Principles (35 minutes)

What are Wireframes? Wireframes are low-fidelity visual representations of your website's structure and functionality. They focus on layout, navigation, and content hierarchy without getting distracted by colors or detailed design.

Types of Wireframes:

- **Low-fidelity:** Basic shapes and placeholders
- **Mid-fidelity:** More detail, basic interactions
- **High-fidelity:** Detailed layouts, near-final structure

Wireframing Best Practices:

- Start with mobile-first design
- Focus on content hierarchy and user flow
- Use consistent spacing and alignment
- Include key interactive elements
- Annotate important functionality
- Keep it simple and functional

Essential Wireframe Elements for Nonprofits:

- Clear navigation with donation CTA
- Impact/mission statement prominently displayed
- Social proof (testimonials, statistics)
- Multiple ways to get involved
- Contact information and transparency links

Practical Activities

Activity 1: Persona Development Workshop (40 minutes)

Students work in pairs to:

1. Choose one Ransem Foundation stakeholder group
2. Conduct mini "interviews" with classmates who role-play as that stakeholder
3. Create a detailed persona using the provided template

4. Present personas to class for feedback and validation

Activity 2: User Journey Mapping Exercise (35 minutes)

Working with their personas, students will:

1. Map the complete journey for their chosen user type
2. Identify emotional highs and lows throughout the journey
3. Highlight pain points and opportunities for improvement
4. Create a visual journey map using sticky notes or digital tools

Activity 3: Wireframe Creation Challenge (50 minutes)

Students use Figma or Canva to:

1. Create wireframes for 3 key pages: Homepage, Donation Page, Volunteer Opportunities
2. Ensure wireframes address their persona's needs and journey
3. Include mobile and desktop versions
4. Add annotations explaining key design decisions

How to Use AI (Claude/ChatGPT) in This Session

AI Prompt Templates for Design & Wireframing

1. Persona Generation:

"Help me create detailed user personas for a nonprofit foundation website. The foundation focuses on [specific cause]. I need personas for:

1. Regular donors (age 35-55)
2. Young volunteers (age 18-30)
3. Beneficiaries seeking services

For each persona, include: demographics, goals, pain points, technology comfort level, and a representative quote. Make them realistic and specific."

2. User Journey Optimization:

"I've mapped this user journey for a nonprofit website: [describe journey steps]. Help me:

1. Identify potential pain points or friction

2. Suggest improvements at each stage

3. Recommend additional touchpoints that could enhance the experience

4. Consider accessibility and mobile-first design principles"

3. Wireframe Validation:

"I'm wireframing a nonprofit foundation website. Here's my homepage layout: [describe layout]. Based on nonprofit best practices, please:

1. Evaluate if this layout effectively communicates trust and impact

2. Suggest improvements for conversion optimization

3. Check if I'm missing any essential nonprofit website elements

4. Recommend mobile responsiveness considerations"

AI Best Practices for Design Sessions

- Provide specific context about your user research findings
- Ask AI to explain the reasoning behind design suggestions
- Use AI to generate multiple design alternatives for comparison
- Validate AI suggestions against real nonprofit website examples
- Remember: AI provides suggestions, but you make the final design decisions

Recommended Tools/Resources

Design & Wireframing Tools

- **Figma** (Free): Professional wireframing and design
- **Canva** (Free tier): Quick wireframes and persona templates
- **Miro** (Free tier): User journey mapping and collaboration
- **Balsamiq** (Trial): Rapid wireframing tool

Learning Resources

- **Video:** "Design Thinking Process Explained" - IDEO (18 min)
- **Article:** "Creating Personas for Nonprofit Websites" - Nonprofit Marketing Guide
- **Template:** User Journey Mapping Canvas
- **Inspiration:** Nonprofit Website Gallery

Research Tools

- **Google Analytics:** Understanding current user behavior
- **Hotjar/Crazy Egg:** User session recordings and heatmaps
- **SurveyMonkey:** Stakeholder and user feedback collection

Mini Quiz - Session 2 Self-Check

1. **Ordering:** Arrange the Design Thinking stages in correct order:
    - Test, Ideate, Empathize, Define, Prototype
2. **Scenario:** You're creating a persona for elderly donors (65+) to your foundation. List 5 specific characteristics this persona should include.
3. **Multiple Choice:** What's the primary purpose of wireframes?
    - a) Show final visual design and colors
    - b) Focus on layout, navigation, and content hierarchy ✓
    - c) Display detailed animations and interactions
    - d) Present marketing copy and brand messaging
4. **Short Answer:** Describe two pain points a first-time volunteer might experience on a nonprofit website and suggest solutions.

Session 3: AI-Powered Development Setup

Learning Objectives

By the end of this session, students will be able to:

- Set up a complete modern web development environment
- Configure and effectively use AI coding assistants (GitHub Copilot, Claude, ChatGPT)
- Initialize and structure a professional web development project
- Implement version control best practices with Git and GitHub
- Leverage AI tools to scaffold project files and generate boilerplate code

Detailed Content Outline

1. Modern Development Environment Setup (30 minutes)

Essential Development Tools:

Code Editor: Visual Studio Code

- Why VS Code: Free, extensible, excellent Git integration, AI assistant support
- Essential Extensions for Web Development:
    - Live Server (real-time preview)
    - Prettier (code formatting)
    - Auto Rename Tag (HTML efficiency)
    - GitHub Copilot (AI assistance)
    - GitLens (enhanced Git capabilities)
    - HTML CSS Support (IntelliSense)

Browser Developer Tools:

- Chrome DevTools for debugging and testing
- Firefox Developer Edition for advanced CSS features
- Browser extensions: Web Developer, ColorZilla, Lighthouse

Command Line Interface:

- Windows: PowerShell or Git Bash
- Mac: Terminal (Zsh)
- Basic commands every developer needs:
    - cd (change directory)
    - ls/dir (list files)
    - mkdir (create folder)
    - git commands (we'll cover these)

2. Version Control with Git & GitHub (35 minutes)

Why Version Control Matters:

- Track changes and project history
- Collaborate safely with team members
- Backup your work in the cloud
- Professional development standard

Git Fundamentals:

```
# Initialize a repository
git init

# Check status of your files
git status

# Add files to staging area
git add .
git add filename.html

# Commit changes with message
git commit -m "Add homepage structure"

# Connect to GitHub repository
git remote add origin https://github.com/username/ransem-foundation.git

# Push changes to GitHub
git push origin main
```

GitHub Best Practices:

- Use descriptive commit messages
- Commit small, logical changes frequently
- Always pull before pushing in team environments
- Use branches for new features (advanced topic for later weeks)

Professional Repository Structure:

```
ransem-foundation/
├── README.md
├── index.html
├── css/
│   └── styles.css
```

```
├── js/
│   └── script.js
├── images/
│   └── (foundation photos/logos)
├── docs/
│   ├── requirements.md
│   └── wireframes/
└── .gitignore
```

3. AI Coding Tools Integration (25 minutes)

Types of AI Development Assistants:

1. Integrated Coding Assistants:

- **GitHub Copilot:** Real-time code suggestions in your editor
- **Tabnine:** AI code completion
- **Replit Ghostwriter:** Cloud-based AI assistance

2. Conversational AI for Development:

- **Claude:** Complex problem-solving, architecture advice, code review
- **ChatGPT:** Quick solutions, debugging help, learning new concepts
- **Bard/Gemini:** Alternative perspectives and solutions

Setting Up GitHub Copilot:

1. Install Copilot extension in VS Code
2. Sign in with GitHub account (student accounts get free access)
3. Configure settings for your coding style
4. Practice with simple HTML/CSS to understand suggestions

AI Coding Best Practices:

- Always understand the code before using it
- Test AI-generated code thoroughly
- Use AI for learning, not just copying
- Combine AI suggestions with your own problem-solving

4. Project Scaffolding and File Structure (20 minutes)

Professional Project Organization:

- Separate concerns: HTML structure, CSS styling, JavaScript behavior
- Use semantic file and folder names
- Include documentation from day one
- Set up development vs. production environments

Foundation Website File Structure:

```
ransem-foundation/
├── index.html            (Homepage)
├── about.html            (About the foundation)
├── programs.html         (Services and programs)
├── donate.html           (Donation page)
├── volunteer.html        (Volunteer opportunities)
├── contact.html          (Contact information)
├── css/
│   ├── reset.css         (CSS reset/normalize)
│   ├── main.css          (Main styles)
│   └── responsive.css    (Mobile styles)
├── js/
│   ├── main.js           (General functionality)
│   └── donation.js       (Donation form handling)
├── images/
│   ├── logo/
│   ├── heroes/           (Large banner images)
│   ├── programs/         (Program photos)
│   └── team/             (Staff/volunteer photos)
└── assets/
    ├── documents/        (PDFs, reports)
    └── icons/            (SVG icons)
```

Practical Activities

Activity 1: Complete Environment Setup (45 minutes)

Students will:

1. Install VS Code and essential extensions
2. Set up Git on their local machine
3. Create GitHub account and first repository
4. Configure GitHub Copilot (if available)
5. Test the complete setup with a simple "Hello World" project

Activity 2: Ransem Foundation Project Initialization (40 minutes)

Working individually, students will:

1. Create the complete folder structure for the Ransem Foundation website
2. Initialize Git repository and connect to GitHub
3. Use AI tools to generate initial boilerplate files
4. Create a professional README.md file
5. Make their first commit and push to GitHub

Activity 3: AI-Assisted Code Generation Workshop (35 minutes)

Students practice using AI tools to:

1. Generate HTML boilerplate for the homepage
2. Create CSS reset/base styles
3. Write JavaScript for basic form validation
4. Generate responsive navigation menu code
5. Create sample content that matches the foundation's mission

How to Use AI (Claude/ChatGPT) in This Session

AI Prompt Templates for Development Setup

1. Project Structure Generation:

"I'm building a website for a nonprofit foundation that focuses on [cause]. Help me create:

1. A professional folder structure for the project
2. A list of essential HTML pages needed
3. CSS file organization recommendations

4. A detailed README.md template that includes project description, setup instructions, and contribution guidelines"

2. Boilerplate Code Generation:

"Generate starter HTML boilerplate for a nonprofit foundation website homepage. Include:

- Semantic HTML5 structure
- Meta tags for SEO and accessibility
- Responsive viewport settings
- Placeholder sections for: hero banner, mission statement, programs overview, donation CTA, footer
- Comments explaining each section's purpose"

3. Git Workflow Assistance:

"I'm new to Git and GitHub. Help me understand:

1. The essential Git commands for solo web development
2. How to write good commit messages for a website project
3. What should be included in a .gitignore file for web development
4. Best practices for organizing commits during development"

4. Development Environment Optimization:

"I'm setting up VS Code for web development focused on HTML, CSS, and JavaScript. Recommend:

1. The most essential extensions for productivity
2. Settings configuration for better coding experience
3. Debugging setup for web development
4. Integration with AI coding assistants"

AI Integration Strategies for This Session

- Use AI to explain complex setup procedures step-by-step
- Generate multiple file structure options and compare them
- Ask AI to review your setup for missing components
- Use AI to troubleshoot installation or configuration issues

- Get explanations of best practices and why they matter

Recommended Tools/Resources

Development Environment

- **Visual Studio Code:** code.visualstudio.com
- **Git:** git-scm.com
- **GitHub:** github.com (Free accounts available)
- **GitHub Desktop:** GUI alternative to command line Git

Learning Resources

- **Video:** "Git and GitHub for Beginners" - freeCodeCamp (1 hour)
- **Interactive:** "Learn Git Branching" - learngitbranching.js.org
- **Cheat Sheet:** Git commands reference card
- **Tutorial:** "VS Code for Web Development" - Microsoft Learn

AI Coding Tools

- **GitHub Copilot:** copilot.github.com (Free for students)
- **Tabnine:** Alternative AI coding assistant
- **Claude/ChatGPT:** For architectural advice and complex problem-solving

Project Management

- **GitHub Projects:** Built-in project management
- **Trello:** Simple task organization
- **Notion:** Documentation and planning hub

Mini Quiz - Session 3 Self-Check

1. **Command Line:** What Git command would you use to save your current changes with the message "Add donation form validation"?
2. **Multiple Choice:** Which file should contain your main homepage content?
    - a) main.html
    - b) home.html
    - c) index.html ✓
    - d) default.html

3. **True/False:** You should commit your code to Git only when your entire website is complete. (False - commit frequently with small, logical changes)
4. **Scenario:** You're using GitHub Copilot and it suggests code you don't understand. What should you do?
    - Research and understand the code before using it
    - Ask AI to explain the code line by line
    - Test the code in a safe environment
    - All of the above ✓
5. **Short Answer:** Name three benefits of using version control for web development projects.

Week 1 Wrap-Up & Next Steps

Key Takeaways from Week 1

- **Requirements gathering** is the foundation of successful web development
- **User-centered design** ensures your website serves real needs
- **Professional development setup** accelerates your productivity
- **AI tools** are powerful assistants, not replacements for understanding

Preparation for Week 2

Students should have:

- [ ] Completed SRS document for Ransem Foundation
- [ ] Finalized user personas and wireframes
- [ ] Fully configured development environment
- [ ] GitHub repository with initial project structure
- [ ] Basic understanding of AI tool integration

Ongoing Project Status

By the end of Week 1, the Ransem Foundation website project should have:

- Clear requirements and stakeholder understanding
- User personas and journey maps

- Wireframes for key pages
- Professional project structure
- Version control setup
- Development environment ready for coding

Next Week Preview: We'll begin actual development with HTML structure and semantic markup, continuing to leverage AI tools for efficient coding and problem-solving.