Revised Self-Report Assessment of Functional Visual Performance (R-SRAFVP) Application

Project Plan

Kirk Hedlich

Dr. Byron DeVries

CIS Master’s Project, Fall 2021

Grand Valley State University

[**Team members**](#_au2zglqfy02z) **3**

[**Development Process**](#_hi7xfeojd6gx) **3**

[**Deliverables**](#_lyy8r1706tfe) **3**

[**Anticipated Technologies**](#_fx2zlurtgvk6) **4**

[**Anticipated Problems**](#_axkj3s26s4tk) **4**

[**Timeline**](#_k7rta33st0f1) **4**

[High Level Plan (Iteration Version 1)](#_sp191qfad7fu) 4

[Detailed Plan (Iteration Version 1)](#_dasyb3xfg8q) 4

# Team members

* Kirk Hedlich
* Dr. Barstow (requirements, design review, final product evaluation)
* Dr. Warren (requirements, design review, final product evaluation)
* Chris Hedlich, OT/L, MS LVSC requirements, design review, final product evaluation)

# Development Process

* Using a simple Iterative development model for the project.
* Using a simple WBS to plan and assign work to the weeks in the semester (see Timeline section)

# Deliverables

* RSRAFVP History
* Project Plan
* Use Cases Specification (Use Case Diagram and Use Cases)
* Requirements Specification
* Requirements Traceability Matrix
* Activity Diagrams Specification
* UI/UX Design Specification
* System Architecture Specification
* Class diagram (this depends on tech and cloud)
* Code under version control (Github)
* Test Cases Document
* Final project release
* Master’s project report
* Master’s project presentation

# 

# Anticipated Technologies

* Google productivity suite (docs, sheets, slides, etc.)
* Github (version control and repository for all things RSRAFVP related)
* Python language and libraries
* PWA using Firebase and Google Cloud

# Anticipated Problems

* Under-planning the number and types of assignments
* Being too optimistic for assignments to get the work done
* Technical challenges with tools, technology or development
* Scope creep to improve functionality and capability before delivering a working tool
* Too many assignments to cover in semester, what could be cut out.

# Timeline

## High Level Plan (Iteration Version 1)

* Planning/Definition (weeks 1-3): Complete the project plan, RSRAFVP history, use cases, and requirements deliverables.
* Design (weeks 4-6): Complete the activity diagrams for user flows, start the traceability matrix with mapping requirements to initial deliverables, develop UI/UX design and user flows, and the initial work for prototyping of the PWA, potentially testing technologies on desktop, tablet and phone devices.
* Development (weeks 7-12): Complete the system architecture, the development of the PWA, testing on different platforms for all planned testing types.
* Deploy (weeks 12-13): deploy project to cloud platform for public use and feedback.
* Debug (weeks 12-14): resolve production issues.
* Presentation (weeks 12-14): Complete the project report, project presentation for final submission in week 14.

## Detailed Plan (Iteration Version 1)

Listed are the work assignments and the week they are expected to be delivered by.

* Week 1 (Sept 2)
  + Project Plan (review)
* Week 2 (Sept 9)
  + RSRAFVP History
  + Project Plan
* Week 3 (Sept 16)
  + Use Cases Specification (Use Case Diagram and Use Cases)
  + Requirements Specification
  + Requirements Traceability Matrix (initial draft)
* Week 4 (Sept 23)
* Week 5 (Sept 30)
* Week 6 (Oct 7)
  + Activity Diagrams Specification
  + UI/UX Design Specification
  + Prototyping PWA (initial with service integrations)
* Week 7 (Oct 14)
  + System Architecture Specification
  + Test Cases Document (initial)
* Week 8 (Oct 21)
  + Class diagram (initial)
* Week 9 (Oct 28)
* Week 10 (Nov 4)
* Week 11 (Nov 11)
* Week 12 (Nov 18)
  + Code under version control (Github)
  + Test Cases Document (final with results)
  + Requirements Traceability Matrix (final)
  + PWA (final)
* Week 13 (Nov 25)
* Week 14 (Dec 2)
  + Master’s project report
  + Master’s project presentation
* Week 15 (Dec 9)