

SEPTEMBER 26, 2020

# PROJECT PLAN

TEAM 41: HR SUPPORT APP

PREPARED BY: TEAM 41

CLAY SPEIDEL

CURTIS KOVACS

DAVID WINGARD

PENELOPE BENAVIDEZ

TY FOSTER

## TABLE OF CONTENTS

Project Description	2
Overview (CO-5):	2
Key Requirements (CO-5):	3
Deliverables (CO-5, CO-6):	3
Acronyms and Abbreviations (CO-7):	4
Design and Architecture (CO-1, CO-3)	5
Implementation Strategy	6
High-level Work Breakdown Structure (CO-2):	6
Schedule / Timeline (CO-2):	8
Required Hardware (CO-2):	9
Third party content (CO-2):	9
Quality (CO-2):	9
Other Special considerations (CO-7, CO-3):	10
Process	10
Process Description and Justification (CO-2)	10
Tools (CO-2):	11
Roles and Responsibilities (CO-2):	11
Location of Project Artifacts (CO-2):	12
Sponsor Communications (CO-7):	12
Risk management	12
Identified Potential Risks (CO-2):	12
Mitigation Strategies (CO-2, CO-3):	12

## PROJECT DESCRIPTION

### OVERVIEW (ABET-2):

- This application allows for an assessment of an individual's performance by one or more people across two different contexts- staffing and sales. After an assessment is completed a report is generated identifying the strengths and challenges of the individual and they are placed on a plot to see where their performance lands. The current manual process is 10 years old and the client would like for it to be automated. The application is intended for members of management and will be deployed as a web and mobile application.

### GLOBAL TRENDS (EM@FSE-ε)

- This project is focused on data-driven decision making using an innovative tool and process. As the world moves towards a future dominated by Big Data, innovative approaches such as these become more valuable in an effort to make tough data-driven decisions.

### MARKET ANALYSIS (EM@FSE-κ)

- This product is an "employee performance evaluation" solution. HR teams are trending towards quicker, casual performance reviews on a somewhat regular basis. This product actually falls into that newer trend. This product is quick and easy to use, but will reveal a lot of information about how the employee is performing in their position.
- Most of these reviews are **not** automated as they are mainly hand-written or a series of face-to-face meetings. They typically solve the problem of evaluating an employee's performance using this process:
  - The employee being evaluated writes a self-assessment
  - The manager reviews the assessment and can optionally take feedback from other employees
  - The manager writes their own assessment of the employee
  - Decisions are made regarding the employee's salary, career path, or termination
- As more and more businesses switch to a more remote business model due to a certain pandemic, there is a growing need for a remote asynchronous tool like this one to simplify performance evaluations. This product is less time intensive, remote, and could possibly give even more accurate data due to the nature of the questions and the analysis involved.
- It is difficult to put a dollar amount on the cost of HR companies doing in-person employee evaluations, but since most companies do them with almost every employee the cost must be great. Assuming an average manager and employee spend three hours each on that employee's annual review, that is 6 hours of productivity lost on one review. [\[resource\]](#)
- This product could save a company a lot of time and money due to the fact that it is much less time intensive and the employees never have to schedule meetings to meet in person.

## SECURITY CONSIDERATIONS (SER-2)

- The most sensitive information that the product will hold is login information. If security is breached this information could be used to utilize the product for free, resulting in loss of profits. Any personal data regarding the performance or perceived performance of individuals that would be considered confidential or sensitive in nature may also pose a security concern. The functionality and intellectual property of the scoring system may also be a security concern in the event of a data leak.

## KEY REQUIREMENTS (SER-2):

- REQ-HRSA001: Assessments shall be completed on mobile and web platforms
- REQ-HRSA002: There are two forms of assessments, staffer and sales
- REQ-HRSA003: The results of the assessment shall be sent to the person(s) who completed them
- REQ-HRSA004: The program will store assessments affiliated with the person being assessed
- REQ-HRSA005: The program will have a central repository
- REQ-HRSA006: The program will allow multiple people to assess an individual or group and display those results individually in addition to an average
- REQ-HRSA007: The administrator will be notified of completed assessments
- REQ-HRSA008: The result of an assessment will be a 3-4 page document that contains a mixture of graphical representation and text based results and recommendations on next steps
- REQ-HRSA009: The program shall have a login feature
- REQ-HRSA010: The program will be able to compare results of two employees and plot them on the same graphical representation
- REQ-HRSA011: The program will be able to show an individual's performance over time
- REQ-HRSA012: The program will store and associate data points with an individual such as roles and job titles
- REQ-HRSA013: Results will automatically be generated upon completion of assessment
- REQ-HRSA014: The program will include built-in support for future modification of language

## DELIVERABLES (SER-1):

- D1: Backend logic of Entities / Controllers according to SCM
- D2: Server to store and access assessment results (depends on D1)
- D3: Generation of individual/group result
- D4: Generation of unique and averaged result(s) for individual/group (depends on D3)
- D5: Generation of result from comparison of two individuals (depends on D3)
- D6: Evaluator GUI, Administrator GUI
- D7: Manager GUI, Employee GUI

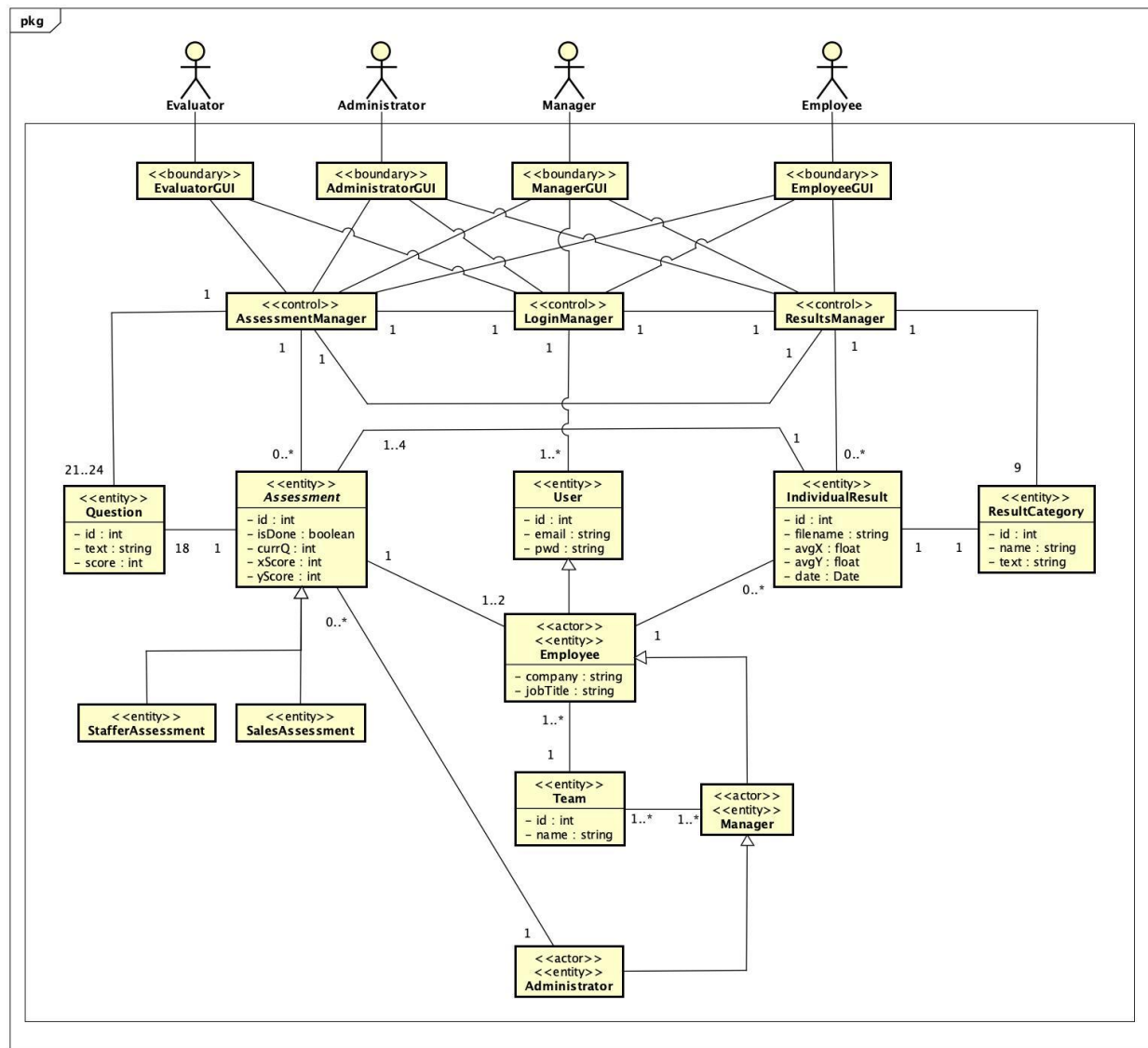
- D8: Communication between server and UIs (depends on D2, D6, D7)
- D9: Generation of individual/group result over time (depends on D2, D3, D6, D7)
- D10: Notification to administrator and email of results to assessor (depends on D2, D3)

ACRONYMS AND ABBREVIATIONS (ABET-3):

- DiSC profile - Dominance, influence, Steadiness, Conscientiousness. Used for discussion of behavioral differences.
- CRM - Customer Relationship Management. A technology that manages customer relationships.
- US - User Stories. Used to represent single user driven goals and tasks.
- UML - Unified Modeling Language. Used to model design and structure of software.

## DESIGN AND ARCHITECTURE

## DESIGN DESCRIPTION (ABET-1, ABET-2)



- The system will need to allow for different users to have access to different types of information, thus, a Model-View-Controller design has been selected. This allows the system to meet REQ-HRSA005.
- The system meets REQ-HRSA002 by having a superclass Assessment for subclasses of specific types of assessments.
- The LoginManager and User classes allow for REQ-HRSA009 to be met.
- The Employee class will store Employee occupation specific information to meet REQ-HRSA012.
- An IndividualResult will be associated with Assessments so that REQ-HRSA006 can be met as well as REQ-HRSA013.

- ResultCategory and Question exist so that individual Questions or ResultCategorys can be modified without editing an entire Assessment.
- An Employee will be associated with multiple IndividualResults and these IndividualResults will have a date attached to them, this will help the system meet REQ-HRSA011 and REQ-HRSA004
- A Team will exist as a collection of Employee's to allow for REQ-HRSA010 to be met.
- Assessments are associated with an Administrator to help meet REQ-HRSA007
- The ResultCategory contains the text for the generated document for an Employee's result, this makes the system modular while meeting REQ-HRSA008. This requirement is also the basis for IndividualResult containing filename, the path to the file associated with the IndividualResult.
- A Factory design pattern will likely be used to create Assessments and IndividualResults since the Questions and ResultCategorys are their own classes that will be different for different Assessments and IndividualResults.
- The User class exists separate from the Employee class given that there could be a system administrator who manages the Questions and Result categories who would likely not be associated with an IndividualResult, Assessment, nor real world Company using the product.
- The team will use Azure, AWS, or some similar service at the sponsor's discretion to host the application interface and the database. This allows the system to meet REQ-HRSA001 and REQ-HRSA005.

#### ALTERNATE DESIGN POSSIBILITIES (EM@FSE-B)

- An alternative design would be to host a dedicated server as opposed to a shared web server. This alternative would be more costly per-month and this would definitely be at the sponsor's discretion. A benefit of using a dedicated server would be that the server can be configured to meet our specifications.

## IMPLEMENTATION STRATEGY

### HIGH-LEVEL WORK BREAKDOWN STRUCTURE (SER-1):

- D1: Backend logic of Entities / Controllers according to SCM
  - identify attributes, functions and dependencies: knowledge of UML and common design patterns such as the factory pattern, ability to identify necessary attributes and dependencies; expected time 10 days.
  - create entities: knowledge of UML and javascript; expected time 1 week
- D2: Server to store and access assessment results (depends on D1)
  - establishing server: knowledge of AWS, Azure technologies; expected time 1 week
  - construction of organizational system for easy retrieval of data: knowledge of SQL; expected time 2 days.
- D3: Generation of individual/group result
  - implement scoring algorithm for results: knowledge of review scoring process, knowledge of javascript; expected time 1 day.

- generate result based on provided information: needs documentation provided by sponsor and understanding of how the results relate to scores, knowledge of javascript; expected time 3 weeks.
- plot results against self or group results: knowledge of SQL for data retrieval, knowledge of javascript; estimated time 1 week.
- save data assigned to reviewed candidate: knowledge of SQL, javascript; estimated time 2 days.
- send report to appropriate users: knowledge of SQL, javascript, file conversion and formatting; estimated time 1 week.
- D4: Generation of unique and averaged result(s) for individual/group (depends on D3)
  - allow for selection of individuals/groups: knowledge of SQL, javascript; estimated time 1 day.
  - collect relevant data for individual/group: knowledge of SQL, javascript; estimated time 1 day.
  - structure key comparisons: knowledge of SQL, javascript, understanding of key points of comparison; estimated time 1 week.
  - average results for individual/group and display: knowledge of SQL, javascript, understanding key points of comparison and plot display; estimated time 1 week.
- D5: Generation of result from comparison of two individuals (depends on D3)
  - allow for selection of individuals: knowledge of SQL, javascript; estimated time 1 day.
  - collect relevant data: knowledge of SQL, javascript; estimated time 3 days.
  - compare results in key areas: knowledge of SQL, javascript, understanding of key points of comparison; estimated time 1 week.
  - plot results together: knowledge of SQL, javascript, understanding key points of comparison and plot display; estimated time 1 week.
- D6: Evaluator GUI, Administrator GUI
  - create access for different types of reviews: knowledge of javascript and UI design, display flow; estimated time 1 week.
  - construct generated results pages: knowledge of javascript and UI design, review process, display flow; estimated time 2 weeks.
  - construct search system for completed reviews: knowledge of javascript, SQL, and UI design; estimated time 1 week.
- D7: Manager GUI, Employee GUI
  - create access for different types of reviews: knowledge of javascript and UI design, display flow; estimated time 1 week.
  - create flow of review process: knowledge of javascript and UI design, review process, display flow; estimated time 2 weeks.



- construct generated results pages: knowledge of javascript, SQL and UI design, review process and result generation, display flow; estimated time 2 weeks.
- allow for managers to see sent reviews: knowledge of javascript and file conversion and formatting; estimated time 2 days.
- D8: Communication between server and UIs (depends on D2, D6, D7)
  - construct communication: knowledge of server, SQL, javascript, UI design and communication points; estimated time 2 weeks.
- D9: Generation of individual/group result over time (depends on D2, D3, D6, D7)
  - allow for selection of groups or individuals: knowledge of SQL, javascript; estimated time 2 days.
  - display comparisons in key areas: knowledge of SQL, javascript, understanding of key points of comparison; estimated time 1 week.
  - create plot with desired data points: knowledge of SQL, javascript, understanding of key points of comparison and plot design; estimated time 1 week.
- D10: Notification to administrator and email of results to assessor (depends on D2, D3)
  - Compile results: knowledge of javascript, file conversion and formatting; estimated time 1 week.
  - Send and archive results: knowledge of SQL, javascript, file conversion and formatting; estimated time 2 weeks.

#### SCHEDULE / TIMELINE (SER-1):

- Milestones will be two week intervals related to deliverables.
- MS1 10/11: D1 Backend logic of Entities / Controllers according to SCM
- MS2 10/25: D3 Generation of individual/group result
- MS3 11/8: D2 Server to store and access assessment results
- MS4 11/22: D5 Generation of result from comparison of two individuals
- MS5 12/4 (end of semester): Flex time
- MS6 1/24: D4 Generation of unique and averaged result(s) for individual/group
- MS7 2/7: D6 Evaluator GUI, Administrator GUI
- MS8 2/21: D7 Manager GUI, Employee GUI
- MS9 3/7: D8 Communication between server and UIs
- MS10 3/21: D9 Generation of individual/group result over time
- MS11 4/4: D10 Notification to administrator and email of results to assessor
- MS12 4/18: Address other features that sponsor may have come up with
- MS13 4/30 (end of semester): Flex time

**REQUIRED HARDWARE (SER-1, EM@FSE-o):**

- A web server for development and hosting of the project
- Android and iPhone mobile devices for testing of the app, provided by the team

**THIRD PARTY CONTENT (SER-1, EM@FSE-o):**

- iOS Developer License
  - Apple Terms of Use: <https://developer.apple.com/terms/>
- Google Play Store Developer License
  - Google Terms of Use: <https://policies.google.com/terms?hl=en>
- Swiftic License
  - Swiftic Terms of Use:  
<http://app4mobilebiz.wpengine.com/swiftic-website-terms-of-use.html>

**QUALITY (SER-2):**

- Functional
 

The product will be usable without any bugs or errors preventing the user from using the application. Errors and bugs detected will be logged and removed with emphasis on usability. The team will use Javascript and maintain consistent java documentation for maintenance and scalability in the future. The design and development process will be well documented to allow for future developers to understand the structure and process that built the application and aid in it's long term maintainability.
- Non-Functional
 

The product should be cohesive and easy for users to navigate and understand, so as to not interrupt the review process it is meant to aid. Where applicable there will be help and instructions available. The look and feel of the product will be consistent and uncluttered as overseen by the UI designer and approved by the sponsor and development team.
- GitHub Standards
 

Master and development should always be working versions, with all US# branches labeled with the correct User Story number having been branched from development. Commit messages will state the US and task number and adequately describe work done. Commits should be logically grouped and made often to show and save progress in that branch, with pushes to remote repos. When a user story is completed, development will be merged to the current US and tested before pull requests are made. At least one team member must review and approve a pull request. Only the Git Master should merge a pull request to the master branch.
- Testing
 

If applicable a unit test will be created in a user story branch before a US# branch is merged to development. Unit tests will include blackbox and whitebox testing as is appropriate for the unit being tested. Beta testing will be completed by all developers to ensure that the product is

functioning and navigable as desired. Branches will be analysed by and must pass a Travis CI build before being reviewed and merged.

- Code Reviews

Before pull requests are made the developer will complete a self review and include it in the pull request. Before approval a reviewer will also complete a code review checklist and include it in the pull request.

#### REFERENCES/SOURCES OF INFORMATION (EM@FSE-Q).

- Javascript - [[resource link](#)]

Most websites use JS for coding the website's logic

- Building a website using HTML, CSS, and Javascript - [[resource link](#)]

A website can be built using these 3 languages. For this p[roject the team plans on building website with the HR tool embedded before making a working mobile app.

- Using an App Builder - [[resource link](#)]

Worth consideration is the use of an app builder to potentially save time and reuse existing resources created.

#### SCALABILITY (EM@FSE-J)

- The project can be scaled through the use of Azure, AWS, or a similar service that will allow the system to quickly adapt to the number of clients using the system.
- The project will be sustained through the sponsor charging a fee for use of the system, this will cover the costs of the web service used to host the application and database.

#### OTHER SPECIAL CONSIDERATIONS (ABET-2):

- The sponsor and team have identified the possibility of adding features as the team progresses and sees room for improvement in design or functionality and have agreed that the scrum process being used for this project will allow for those changes to be made.
- The sponsor sees room in the future for extending the languages that this project should be available in.

#### PROCESS

##### PROCESS DESCRIPTION AND JUSTIFICATION (SER-1)

- We will be using the Scrum process for development. This will foster incremental progress where each team member will have the opportunity to hold a leadership role. The Scrum process is suitable for the project because it allows for easy translation of sponsor requirements to user stories and allows for changes in design as the team and sponsor work together to form a final product. The use of a Taiga board for tracking the scrum process also allows for team members to work completely asynchronously as each team member is located remotely.

## TOOLS (SER-1, EM@FSE-O):

- jest: JavaScript test suite [[resource link](#)]
- WC3 CSS validation service: validates CSS [[resource link](#)]
- WC3 HTML validation service: validates HTML [[resource link](#)]

## ROLES AND RESPONSIBILITIES (SER-1):

- **Sponsor: Chuck Bean**  
The sponsor is the product owner and is responsible for product conception and direction. The sponsor advises and meets with the development team and provides feedback and approval to make sure the product is progressing as intended.
- **Advisor: Vijay Suthar**  
The Advisor is an Arizona State University staff member responsible for reviewing and assessing the performance and progress of the student team members.
- **Git Master: rotating**  
Git Master is responsible for seeing that the agreed upon Git flow is followed and for approving merges to the master branch. Git Master will be rotated throughout the development process - changing every two sprints, approximately every two weeks.
- **Scrum Master: rotating**  
Scrum Master is responsible for monitoring the scrum process and sprint burn down progress. The Scrum Master will make sure the team is on track to meet sprint goals and help identify and work around barriers to progress. Scrum Master will be rotated throughout the development process- changing every two sprints, approximately every two weeks.
- **Developers: Clay Speidel, Curtis Kovacs, David Wingard, Penelope Benavidez, Ty Foster**  
All developers are responsible for individual contributions to the product. Developers are responsible for identifying, assigning, and completing user stories throughout the development process. Developers will work collaboratively in the design, structure, and creation of the final product.
- **UI Designer: Penelope Beanvidez**  
The UI Designer is responsible for monitoring the cohesion of the products user interface. The UI Designer will work with the sponsor and developers to create a user interface that is consistent and easy to navigate.
- **Task Master: David Wingard**  
The Task Master is responsible for monitoring the progress of the team with respect for class defined deliverables. The Task Master will oversee that group deliverables are submitted on time.
- **Sponsor Contact: Clay Speidel**  
The Sponsor Contact is responsible for email communications with the sponsor, including providing progress updates, receiving and asking questions, and arranging meeting times with the sponsor.

## LOCATION OF PROJECT ARTIFACTS (SER-1):

- To track user stories, sprint progress, and maintain a project backlog the team is using a Taiga scrum board that can be found [here](#). The board will be public to support multiple users and as such will not contain confidential or sensitive information.
- To support an asynchronous programming workflow and collaboration a Github repository will be used and maintained for version management. The repository will be private to limit those who have access and can be found [here](#).
- Documentation organization and management will be on Google Drive for easy access and version control. The Drive file will be restricted to those manually added and can be found [here](#).

## SPONSOR COMMUNICATIONS (ABET-3):

- The team will meet with the sponsor, Chuck Bean, via Zoom meetings in two week intervals on Friday at 2pm MST, 10/9, 10/23, 11/6 and 11/20.
- For the spring semester the team intends to continue a two week interval between meetings and will reassess dates and times to correlate with the pending schedule for SER 402.

## RISK MANAGEMENT

## IDENTIFIED POTENTIAL RISKS (SER-2):

- iOS app development may be hindered by team member access to Apple products and by extension xcode development tools.
- A possible personnel shortage may occur if a team member has any issue restricting their continued enrollment in either SER 401 or SER 402, as those two courses span the length of this project.
- As is common with remote school/work/project scheduling it may be difficult to schedule or keep to scheduled meetings, possibly putting development behind schedule.
- If the project runs behind schedule it is possible that not all requirements can be completed on time.
- It is possible that with a new team there are problems with communication and reliability that can hinder the teams progress.
- There is the possibility of lost work with the use of remote repositories and cloud based documentation such as in the teams use of Google Drive, GitHub, and Taiga.

## MITIGATION STRATEGIES (SER-2):

- There are a few rental and cloud based solutions available to remotely access Apple products and xcode development tools. At least one team member also owns a Macbook and will be able to access xcode.

- In the event that a team member is lost or cannot otherwise participate in the development of the project, the team will work closely with the sponsor and the class advisor to find a reasonable solution or compromise for the loss of the team members projected work.
- In the event of meeting cancelation the team will try it's best to reschedule but will also rely on email communication for questions or concerns that cannot be rescheduled.
- The team has allowed for scheduled over runs but will otherwise work with the sponsor and advisor closely to deliver a comparable solution to any requirements that have not been met to satisfaction.
- In the event that there are irreconcilable issues with work or communication with a team member the team will work closely with the advisor to find a solution and to get the team back on track for project completion.
- The team will keep local versions of remotely stored documents and repositories, updating them after each sprint to lessen the chances of major losses but will also work with the respected team that controls the remote location if any loss does occur.