



MET CS473/673 Software Engineering
7 November 2013

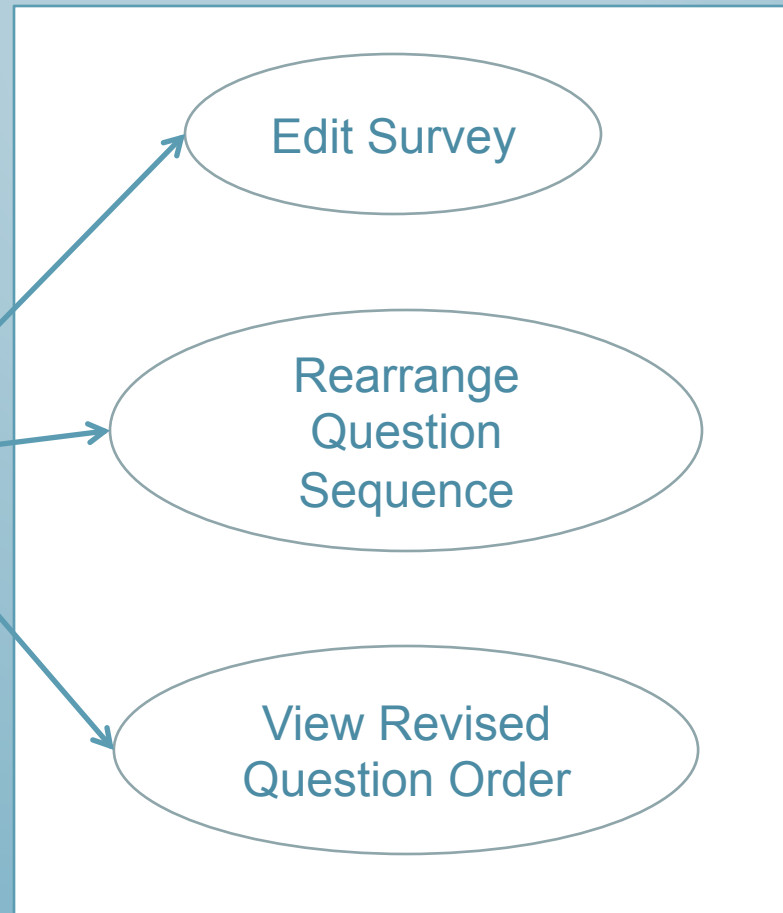
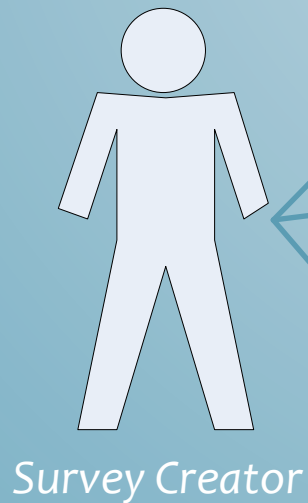
COMPLETED USER STORIES: ITERATION #2

**Change Question
Sequence**

As a survey creator, I should be able to change the sequence and add additional questions at any time before submitting the survey.

USE CASES: ITERATION #2

Change Question
Sequence



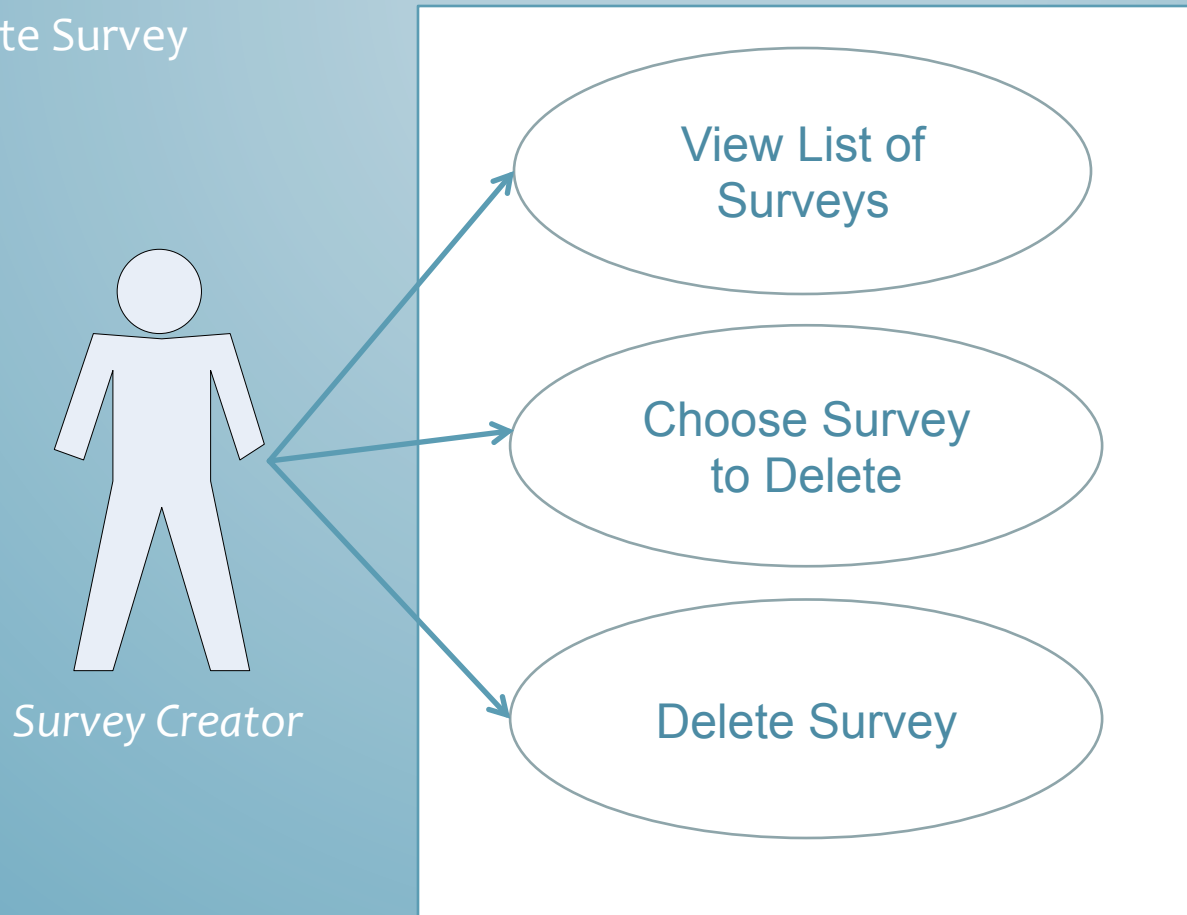
COMPLETED USER STORIES: ITERATION #2

A survey-creator should be able to delete any survey he created. As a survey creator, I can delete any survey that I created so that I can keep the surveys that I have on the site both current and relevant.

Delete Survey

USE CASES: ITERATION #2

Delete Survey



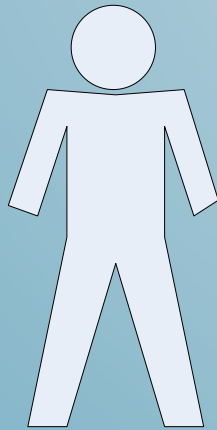
COMPLETED USER STORIES: ITERATION #2

**Publish/Unpublish
Survey**

As a survey creator, I want to be able to publish and unpublish my surveys, because I don't want my survey taken in the event that I find an error and need to fix it.

USE CASES: ITERATION #2

Publish/Unpublish
Surveys



Survey Creator



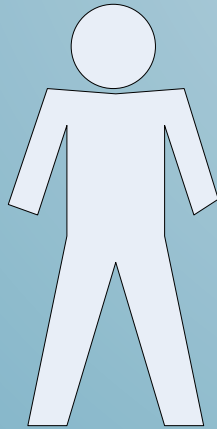
COMPLETED USER STORIES: ITERATION #2

**Home Link
(click on logo)**

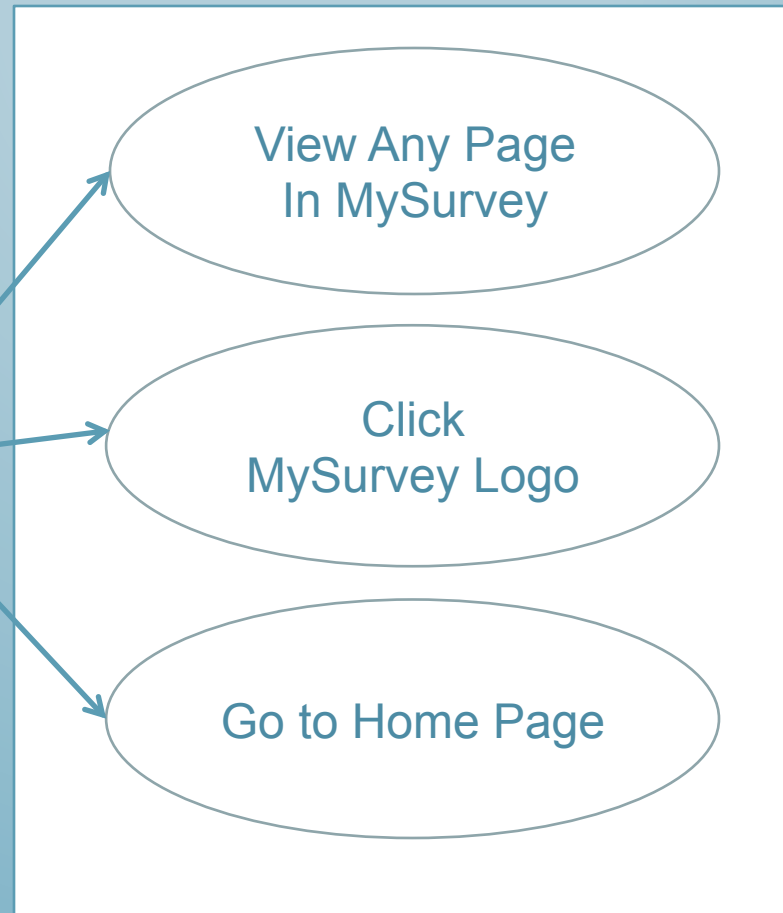
As a survey creator, I would like to return to my survey list when I click the website logo.

USE CASES: ITERATION #2

Home Link
(click on logo)



Survey Creator



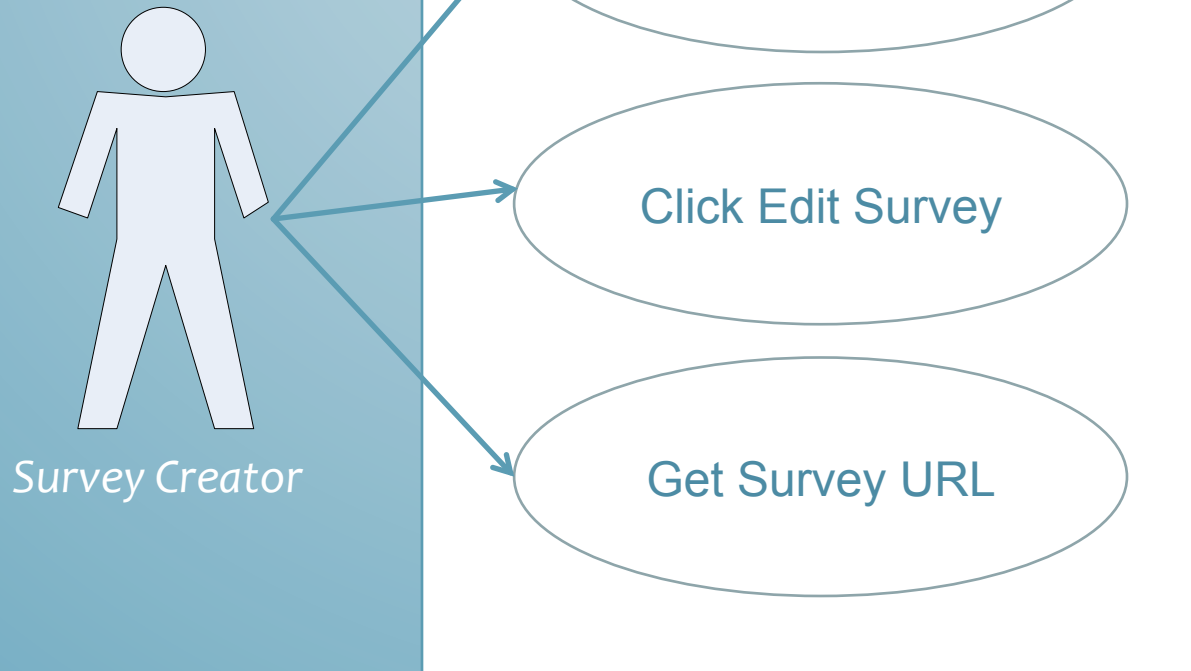
COMPLETED USER STORIES: ITERATION #2

A survey creator should receive a unique URL for each survey he creates. As a survey creator, I should receive a unique URL for each survey I create, because I want to invite people to take my survey.

Survey URL

USE CASES: ITERATION #2

Survey URL



IN PROGRESS USER STORIES: ITERATION #2

As a survey creator I should be able to add different types of questions to the survey.

**Add Different
Types of
Questions**

ADDED USER STORIES: ITERATION #2

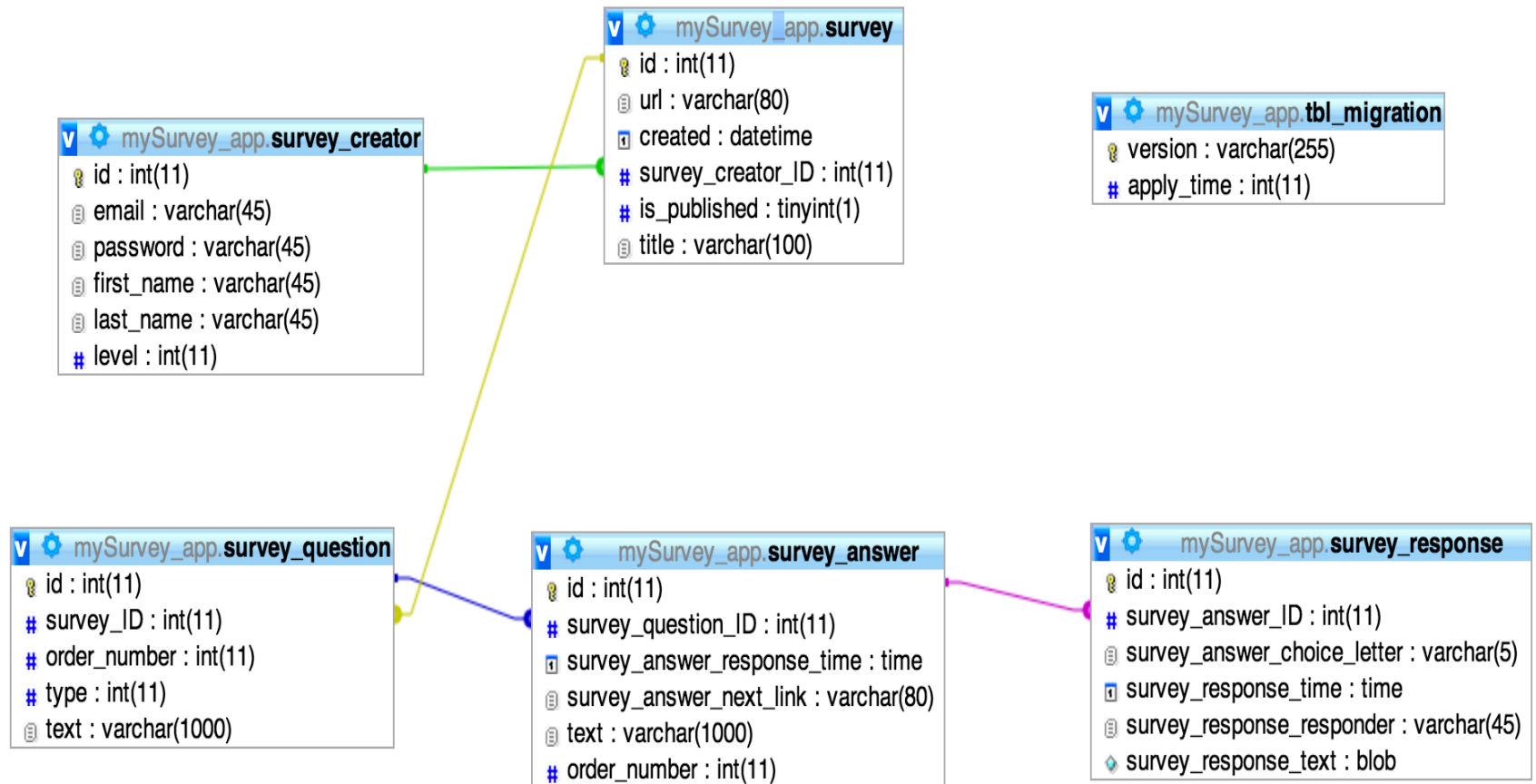
**Survey creator
limits on deleting
and editing survey**

As a survey creator, I should only be able to edit or delete a survey if no one has taken it yet.

CURRENT SOFTWARE ARCHITECTURE

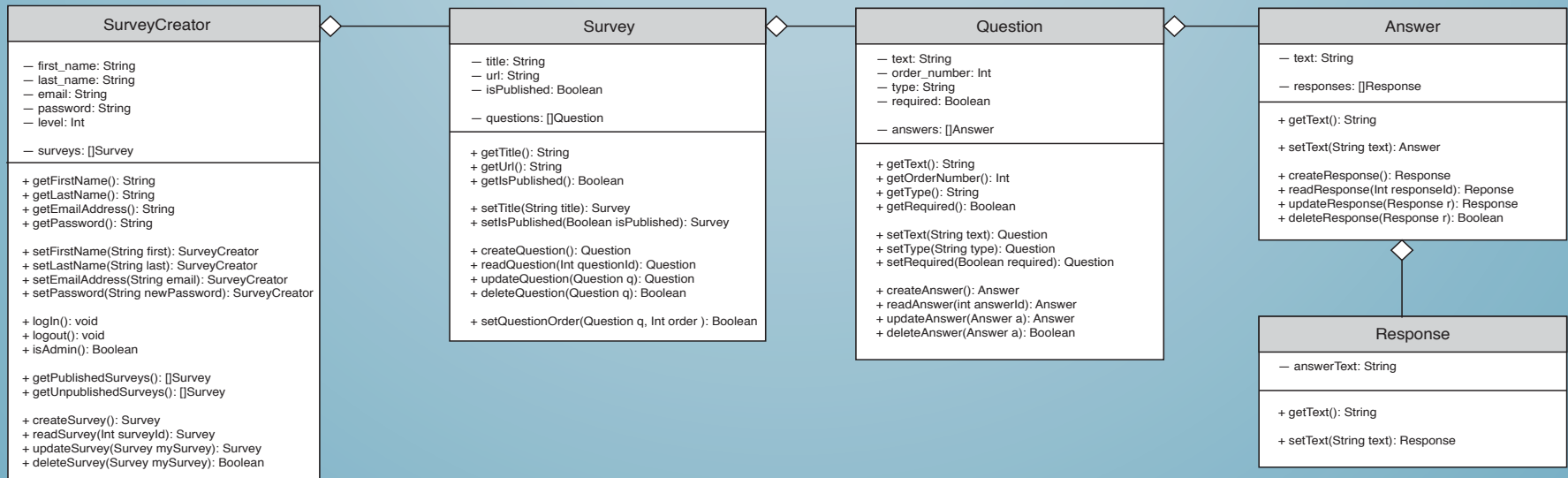
- We are following the MVC architecture
- Every table in our DB has a model, view and a controller
- Model: represents the data and enforces the rules to be validated
- View: renders the pages that the user interacts with (forms, etc.)
- Controller: provides functions to handle UI actions (update, delete, create, etc.)

DATABASE DESIGN



SOFTWARE DESIGN

CLASS DIAGRAM



IMPLEMENTATION

Tools Used:

- Yii PHP framework: provided some base PHP code and helps generate CRUD operations (MVC architecture)
- Apache/php/mysql stack: MAMP, XAAMP, AMPPS
- Github and Sourcetree for version control and issue/bug tracking
- Pivotal tracker for user story management and requirement tracking

IMPLEMENTATION

Features Implemented:

- Re-ordering of questions by dragging questions
- Added 2 types of questions (Simple text and Multiple choice)
- Ability to add/delete answer choices for multiple choice questions
- Unified form
- Added Account Settings (change password, first name, last name)

MYSURVEY APPLICATION

PROJECT DEMO

TESTING

Unit Testing

- *Testing a single function or method of the code*
 - *Ex: The create survey form can be written to the database.*

Functional Testing

- *Testing the whole project by simulating user behavior*
 - *Ex: User can login, create a survey, and logout.*

MYSURVEY APPLICATION

METRICS

Previous
Iteration

6

5

User Stories
Completed

5

6

Bugs
Resolved

4

4

Open
Issues

MYSURVEY APPLICATION

METRICS

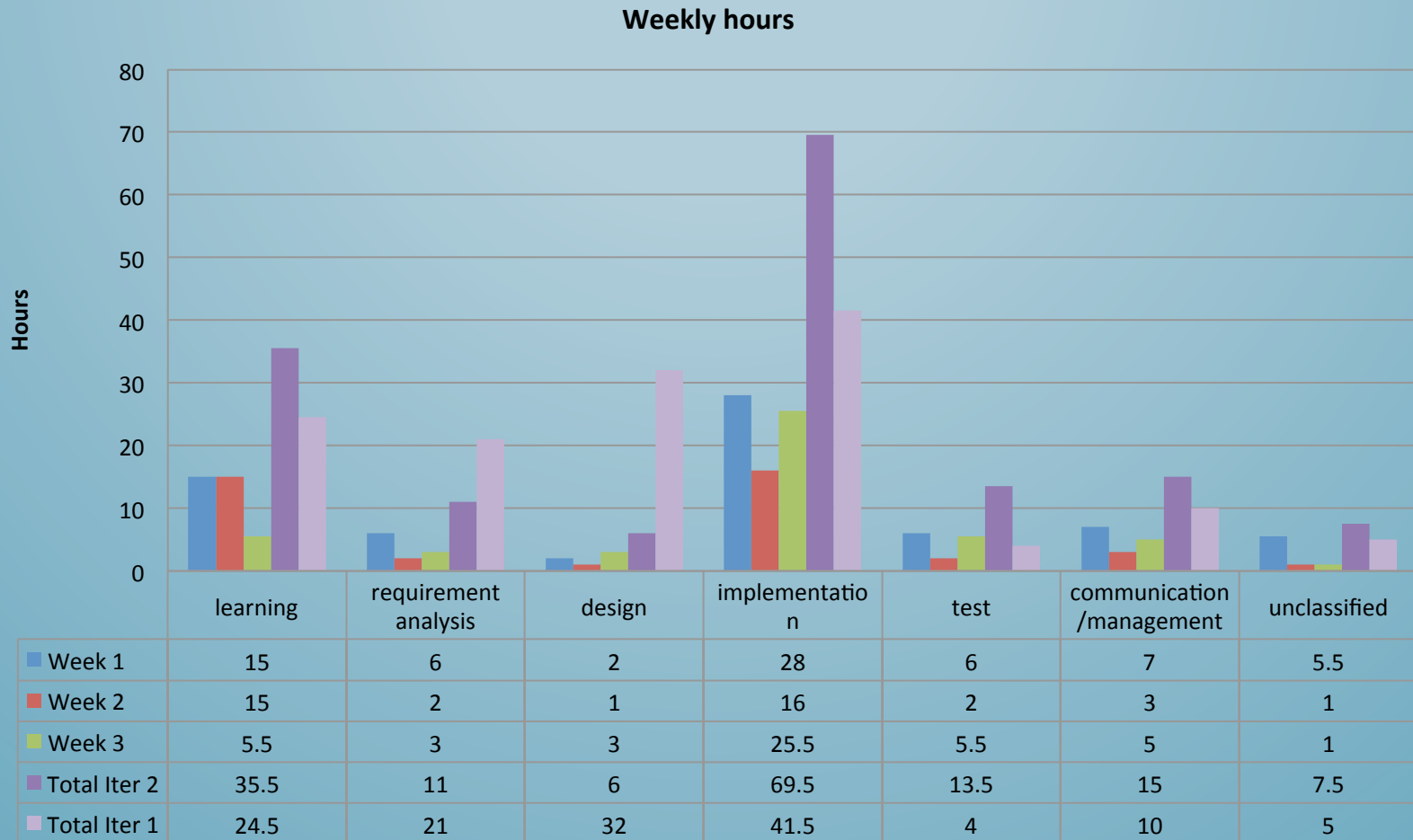
Previous
Iteration

6644

7473

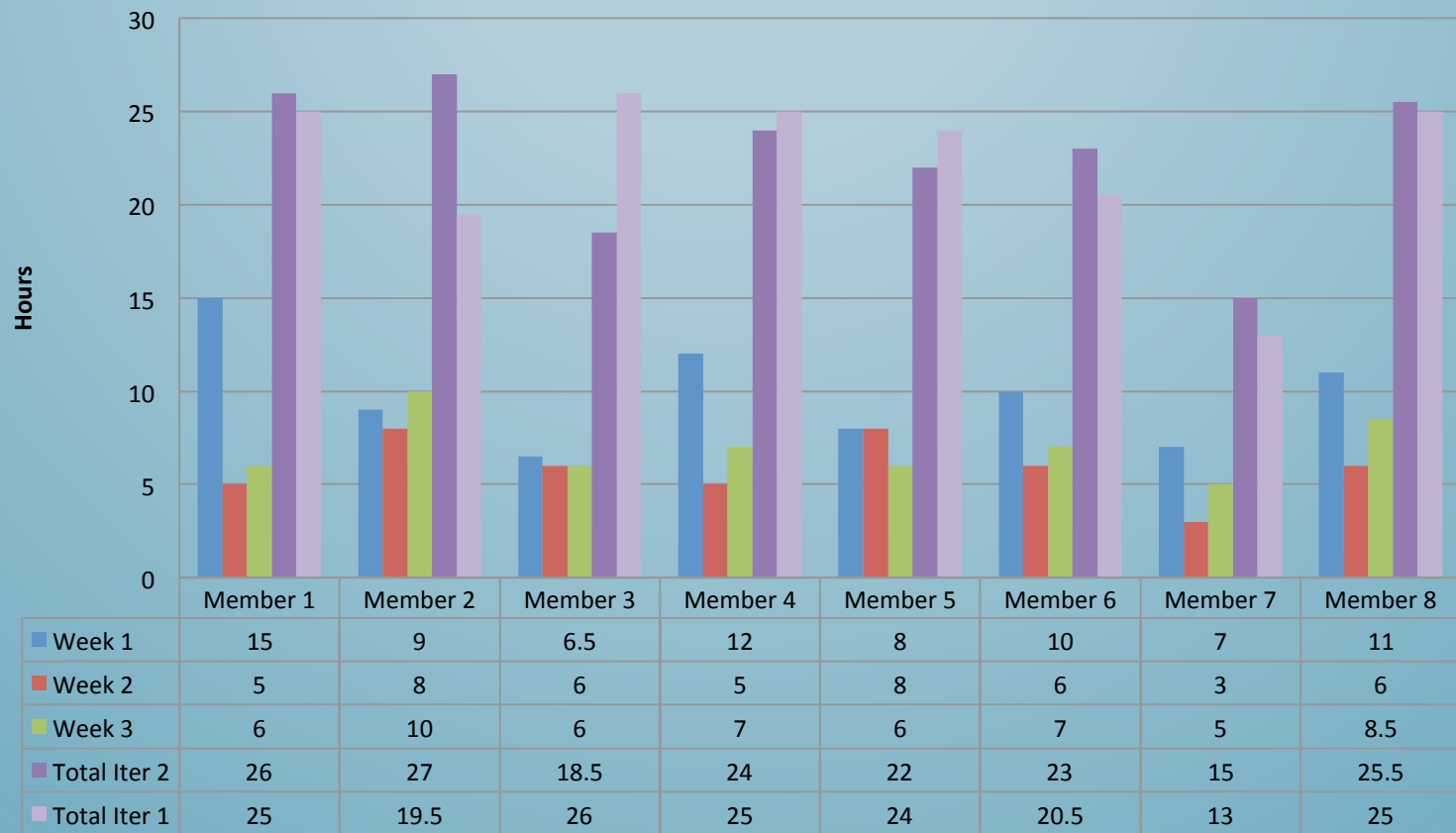
Lines of Code

METRICS



METRICS

Total time for each member



CURRENT PROGRESS

- Database Design, Wireframes & Class Diagram
 - Started in the initial planning phase; ongoing
- Create register, log in & log out functions
 - Programming & unit testing completed in Iteration 1
- Create Survey: survey title, question & answer
 - Programming completed
- Survey question ordering is completed

CURRENT PROGRESS

- Add different types of question to the survey
 - In progress
- Unit Testing Plan
 - Completed
- Functional testing
 - In progress

LESSONS LEARNED

- Daily status updates
 - Every member sends daily status updates to keep every member up to date.
 - It helps members know about potential problems and helps coordinate effort to resolve time-consuming issues.
- Simple and small user stories
 - Split user stories into simple and smaller ones.
- Version consistency
 - Every team member should have the same version of the development tools.
- Estimation
 - Be a “Pessimist” when estimating features and iterations.

FUTURE WORK

- Database Design, Wireframes & Class Diagram
 - Ongoing, will change as requirements change
- Requirements Review
 - Ongoing; may be refined as needed
- Refine survey creation piece
 - Provide more question types
 - Mark question as required
 - Improve flow of survey and questions

FUTURE WORK

- Focus on survey taker part
 - Add functionality to take survey
- Create reports & question sequencing
- Testing & peer code review