MySurvey

mysurvey.net63.net

MET CS473/673 Software Engineering
6 December 2013

PROJECT GOAL

Our goal is to create an internet application that will provide the ability to create, take and analyze surveys.

AUDIENCE + USERS

Survey Creators

- Able to create, edit, and delete surveys
- Can publish or un-publish a survey

Survey Takers

- Accesses a survey from a unique URL
- Can submit answers to survey questions

REQUIREMENTS

analysis

PIVOTAL TRACKER

Sample User Story:

As a survey creator, I should be able to see all my surveys because I want to know what surveys I've created.

Two tasks:

- Design UI
- Code behind

PIVOTAL TRACKER WORKFLOW

- Add features and ideas to the icebox.

- Estimate each feature. Assign difficulty points using a 0, 1, 2, 4, 8 system.

- Move features to the backlog, arrange by highest priority.

- Move features to the current iteration manually or let Pivotal Tracker move them automatically based on the velocity and estimation.

- After completing and testing each feature, mark the user story complete.

MANAGING USER STORIES

- Used Pivotal Tracker comments to record updates
- Agreed on tasks at weekly meetings
- Created new user stories to track functionality changes
- Used Github to record bugs and issues

FUNCTIONAL REQUIREMENTS

Sample Requirement:

A survey creator will register for the site by providing a first name, last name, email address and password. The system will store this data in the database and use a hashing algorithm to store the password.

To login, a survey creator will enter their email address and password, which the system will verify in the database.

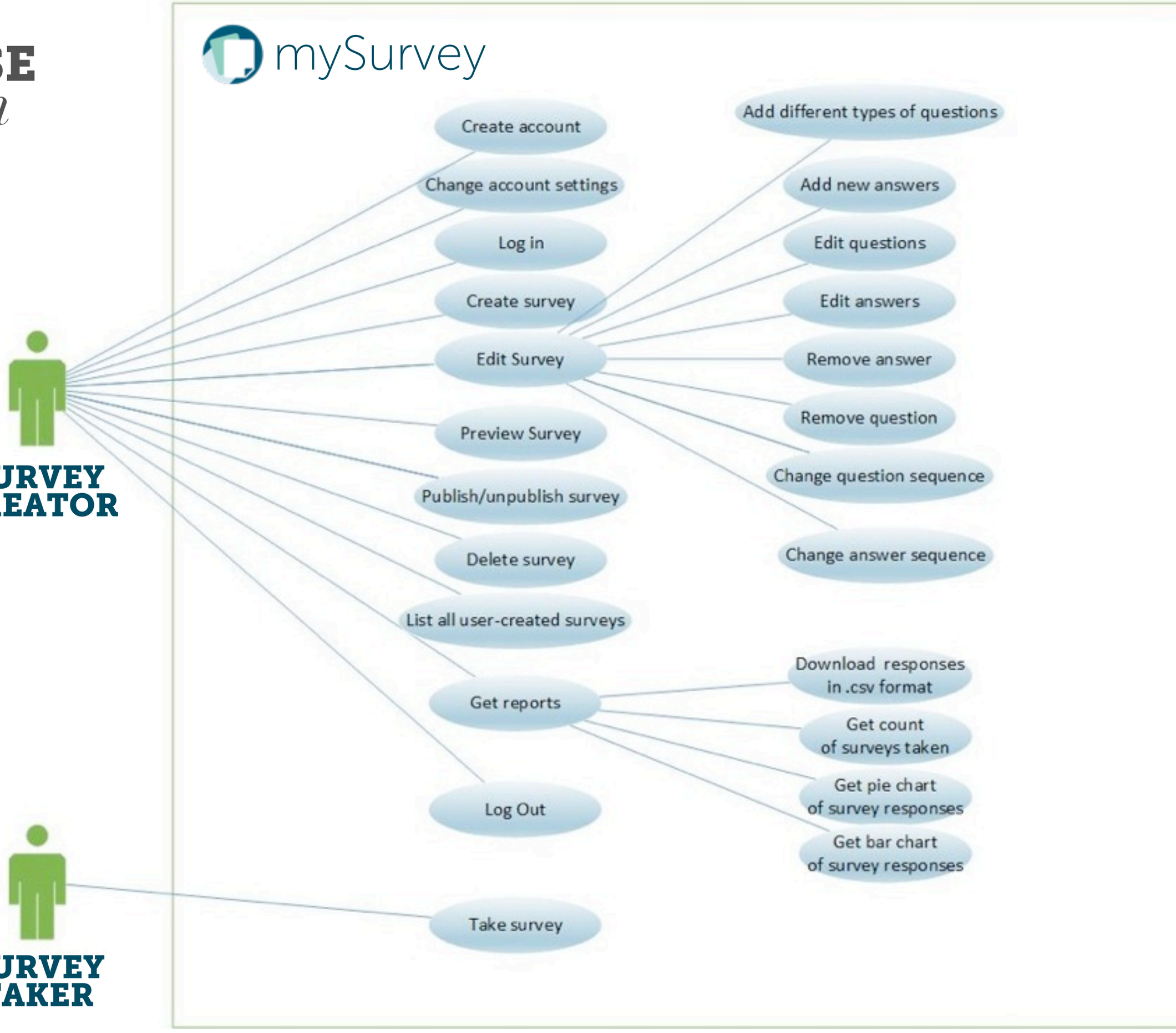
NON-FUNCTIONAL REQUIREMENTS

Sample Requirement:

The system will not collect any personal data from the survey-takers.

Privacy has become a major concern with any online activity. The survey takers must feel free to submit their answers as anonymously as possible, or they will hesitate to take a survey. Because of this, the system will not retain any information from the survey taker other than explicit answers to survey questions.

USE CASE *diagram*



Software
ARCHITECTURE

DEV CONFIGURABLES

- PHP 5.4
- Apache (ModVHost + ModRewrite + short_open_tag=false)
- MYSQL
- Git
- PHPDocumentor
- PHPUnit
- Selenium

YII ARCHITECTURE

- All urls are directed to the index file.
- Desired controller/action for each request are defined by a set of routing rules.
- Actions perform desired logic and render a themed view.

MySurvey Software Architecture (Yii Framework)

user action

CONTROLLER

Defines actions on the views and on the models
Defines filters for data and access rules for views
Loads data from the model for the view

Update state

BROWSER

MODEL

Database interface

Defines rules, relationships and attributes of the data

MySQL Database

VIEW

User Interface

Presents data from the model to the user using HTML and CSS

updated data from model

query state

Based on a model from this site: www2002.org/CDROM/alternate/478 and on MVC diagram in "Head First Design Patterns"

FRAMEWORK NOTABLES

YiiC: CLI support to run project specific commands.

(ie, Documentation, migration, testing)

Migration: Version track database changes

Theme: Interchangeable look and feel for the site.

Config: Different config files for different tasks (CLI vs. Web)

and un-versioned credential files.

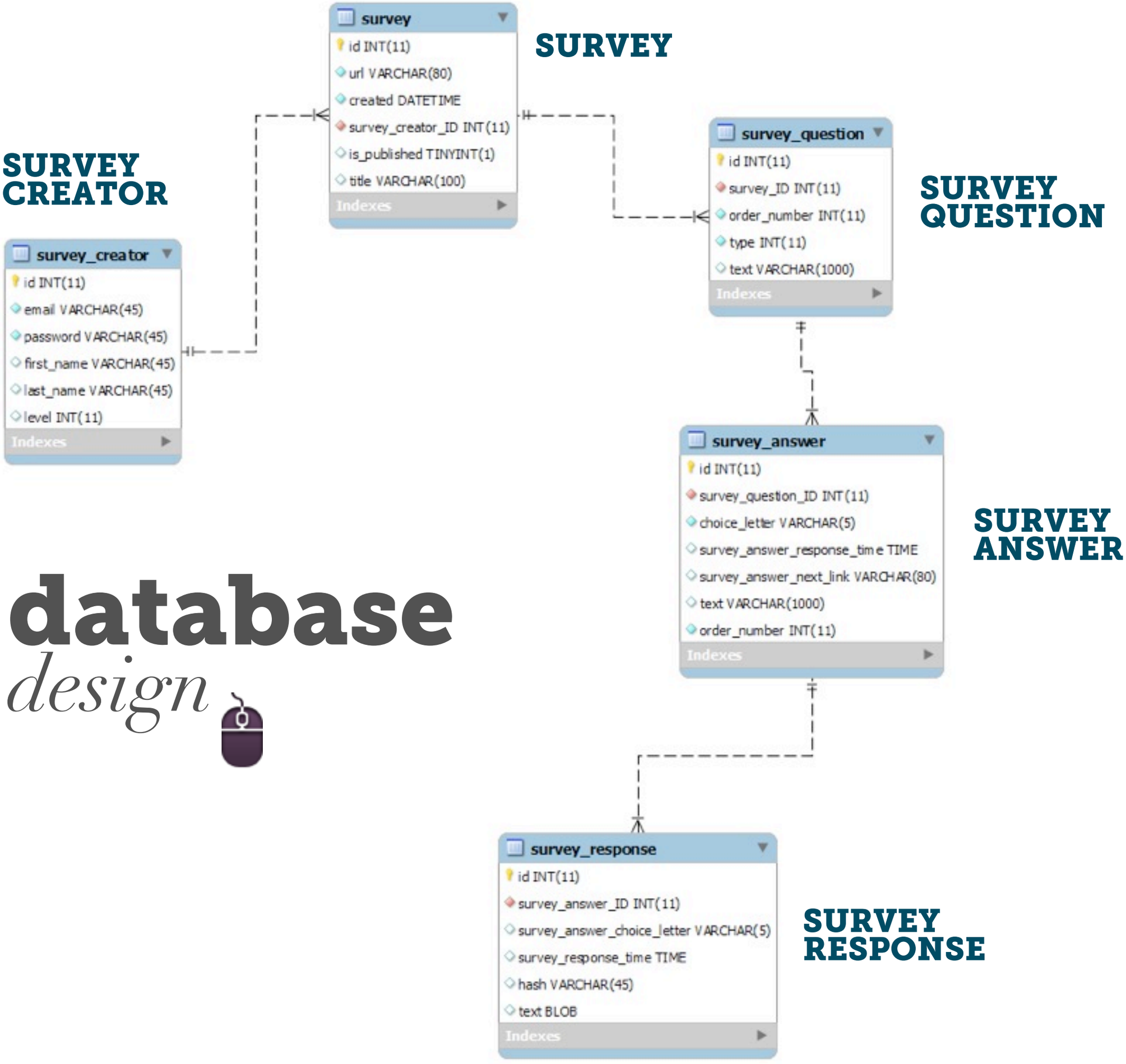
Templates: Reusable envelopes for themed views.

Gii: CRUD generator

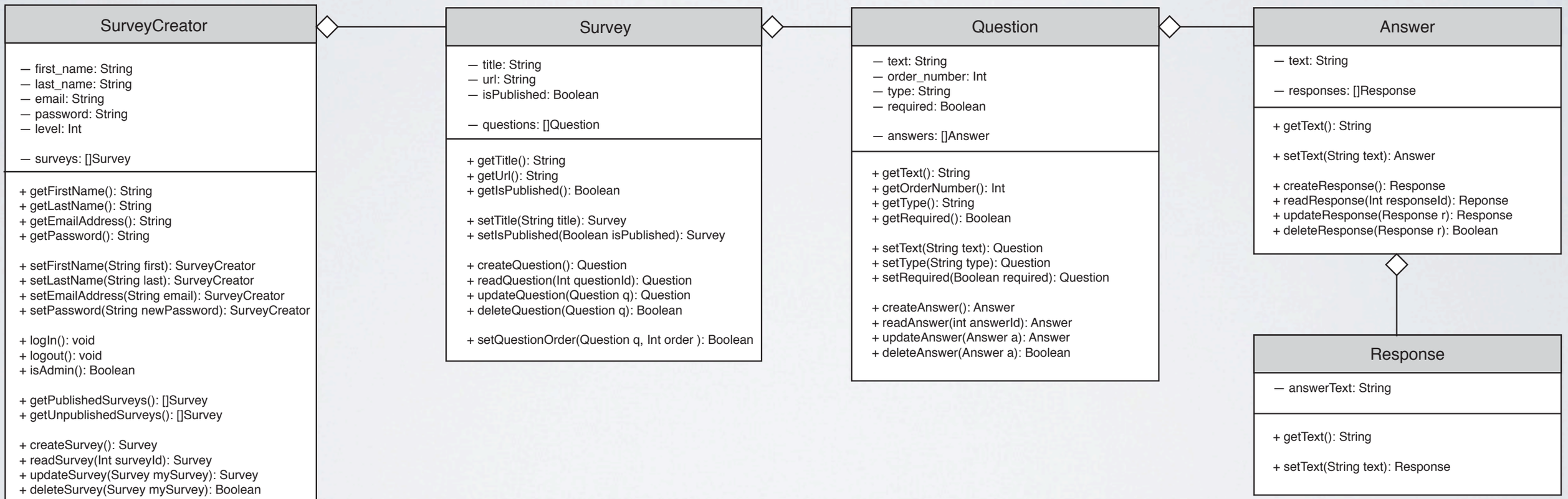
Widgets: Reusable code blocks outside of the controller's scope.

Assets: Runtime cache to improve response time.

database
design



CLASS DIAGRAM



CLASS *relations*

- CComponent
  - |--- CUserIdentity
    - |--- UserIdentity
  - |--- CController
    - |--- Controller
      - |--- AnswerController
      - |--- SurveyController
  - |--- CModel
    - |--- CFormModel
      - |--- ContactForm
      - |--- LoginForm
  - |--- CActiveRecord
    - |--- Model
      - |--- Survey
      - |--- SurveyAnswer
      - |--- SurveyCreator
      - |--- SurveyQuestion
      - |--- SurveyResponse

ALGORITHMS

Unique Hash Generator:

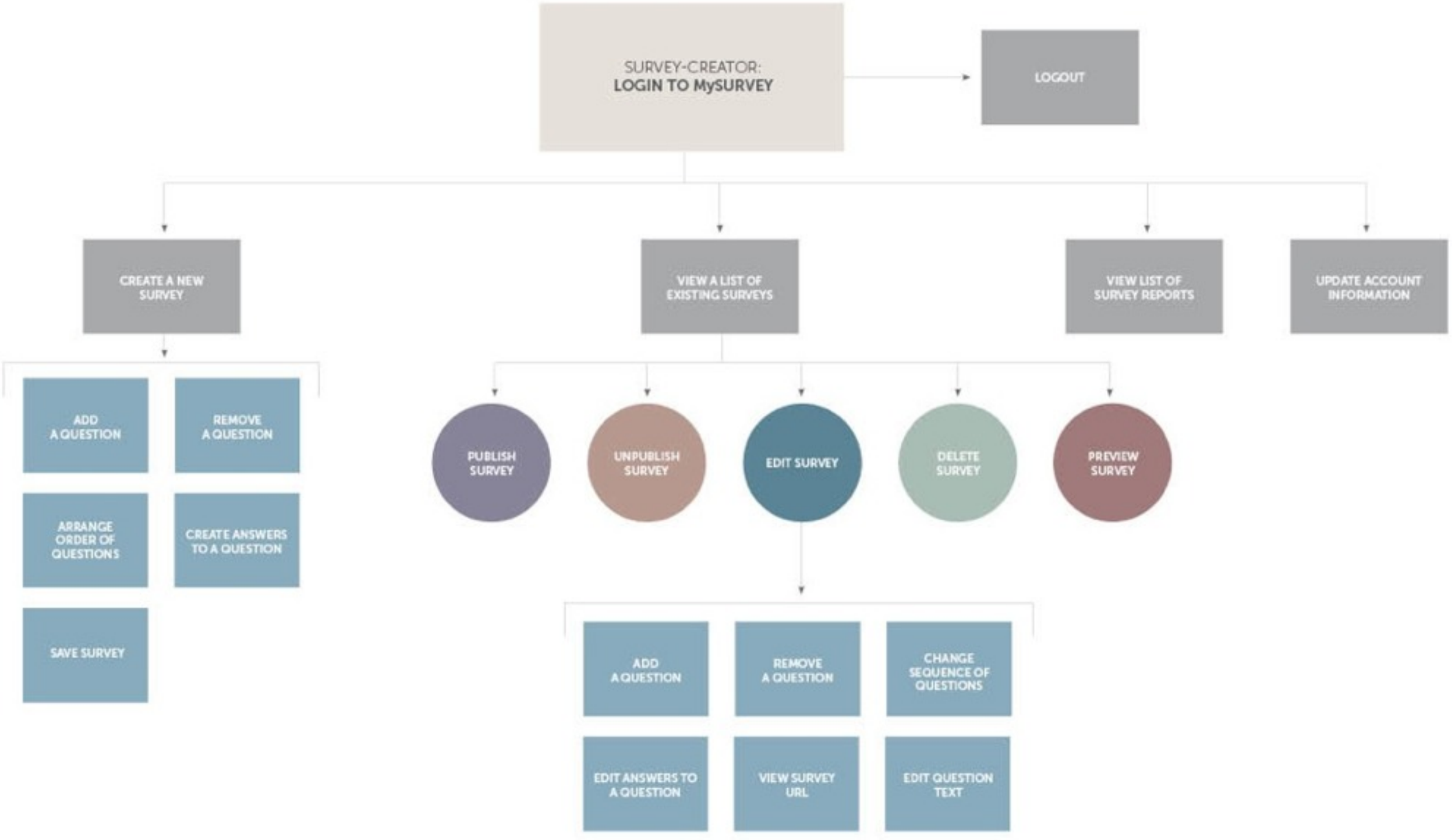
```
private function generate_unique_url($length = 6){
    $result = “”
    for(0->$length)
            $result .= get_random_alpha_num_char();
    if (find_model_with_hash($result))
      $result = generate_unique_url($length);
    return $result;
}
```

ALGORITHMS

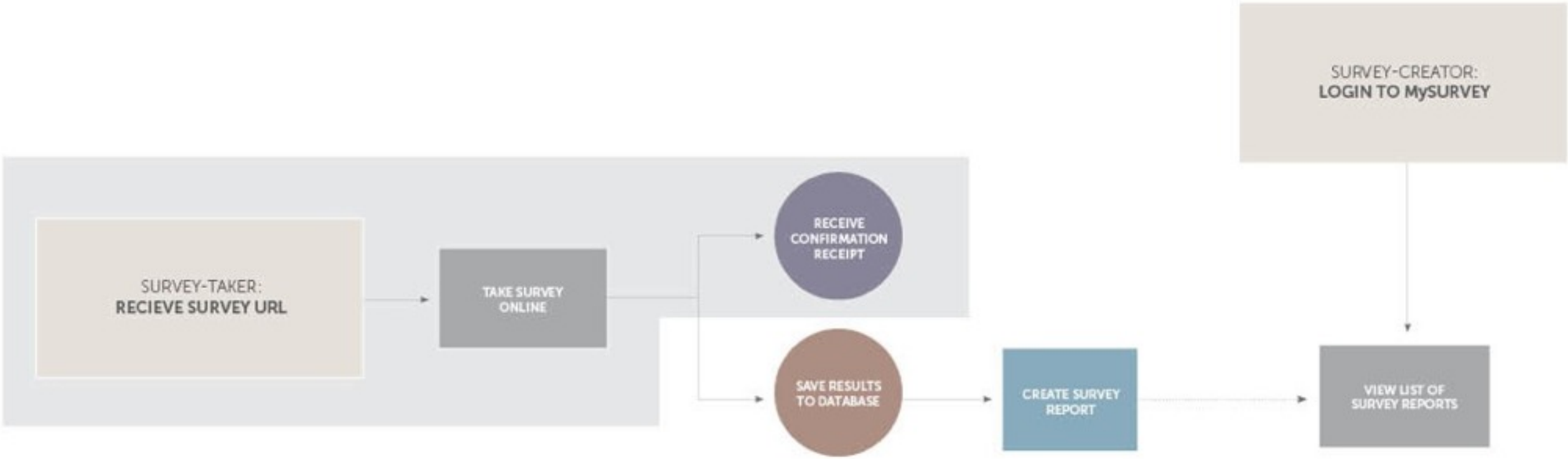
Strategy Pattern: (mySurvey_app/static/js/survey_update.js)

```
strategy = {
    'short-answer':{add:function(){...}},
    'multiple-choice':{add:function(){...}},
    'true-false':{add:function(){...}}
}
onClickAddButton($event)
    elementType = $event.getElementType();
    strategy[elementType].add();
```

design + **UI**


SURVEY-CREATOR:
LOGIN TO MySURVEY
LOGOUT
CREATE A NEW SURVEY
VIEW A LIST OF EXISTING SURVEYS
VIEW LIST OF SURVEY REPORTS
UPDATE ACCOUNT INFORMATION
ADD A QUESTION
REMOVE A QUESTION
ARRANGE ORDER OF QUESTIONS
CREATE ANSWERS TO A QUESTION
SAVE SURVEY
PUBLISH SURVEY
UNPUBLISH SURVEY
EDIT SURVEY
DELETE SURVEY
PREVIEW SURVEY
ADD A QUESTION
REMOVE A QUESTION
CHANGE SEQUENCE OF QUESTIONS
EDIT ANSWERS TO A QUESTION
VIEW SURVEY URL
EDIT QUESTION TEXT

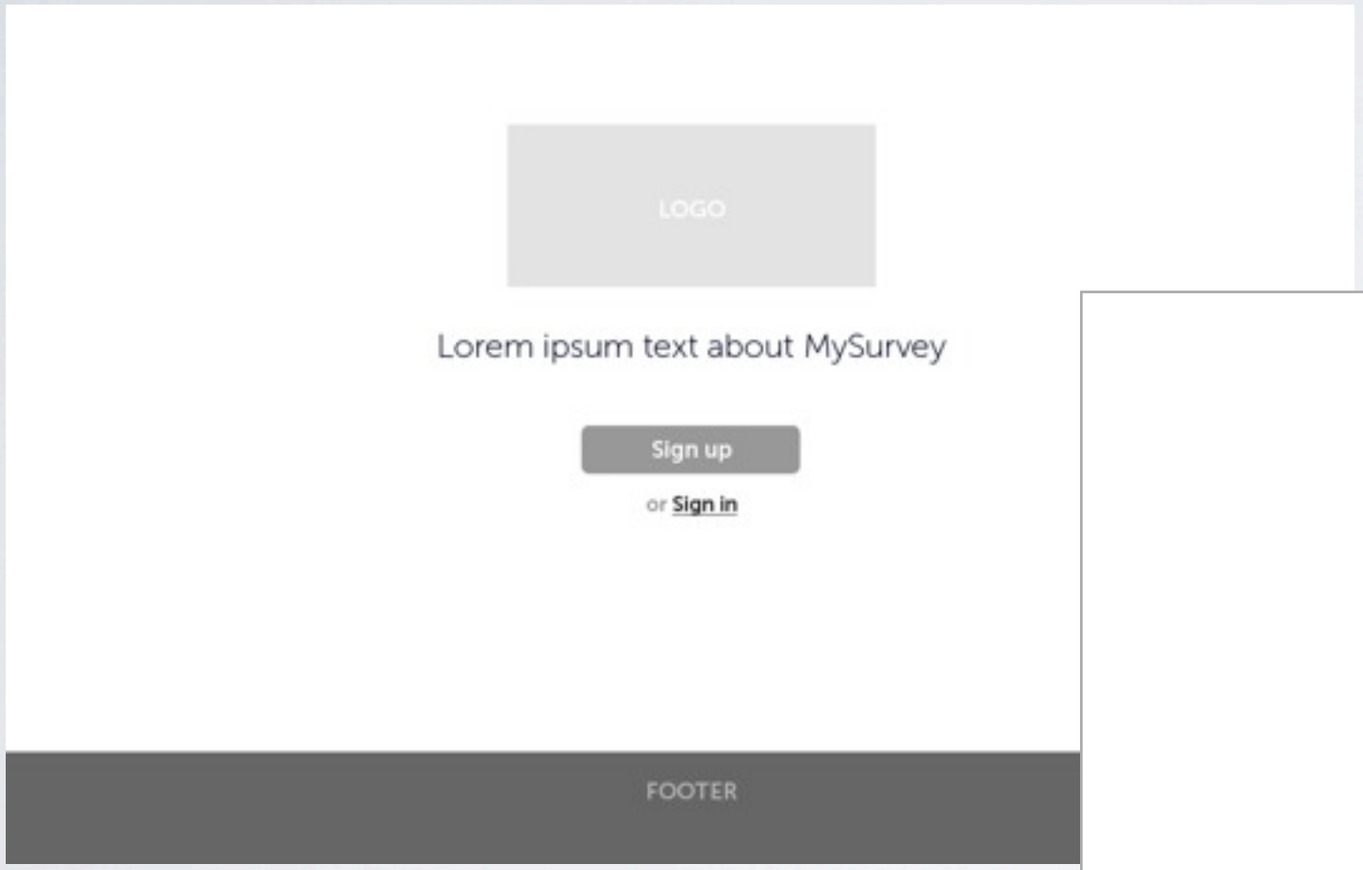
USER *flow*



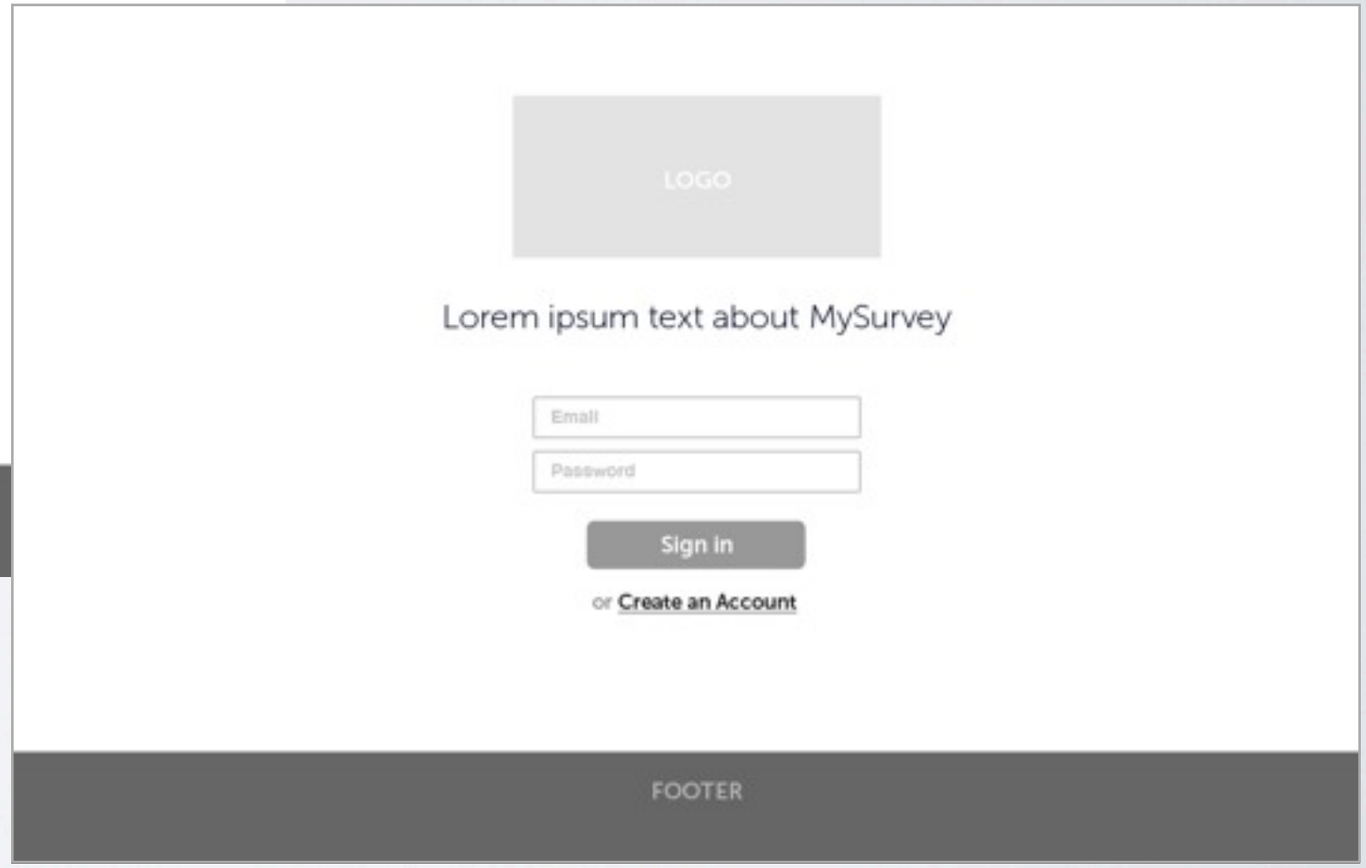
= VIEWABLE TO SURVEY-TAKER

USER *flow*

WIREFRAMES

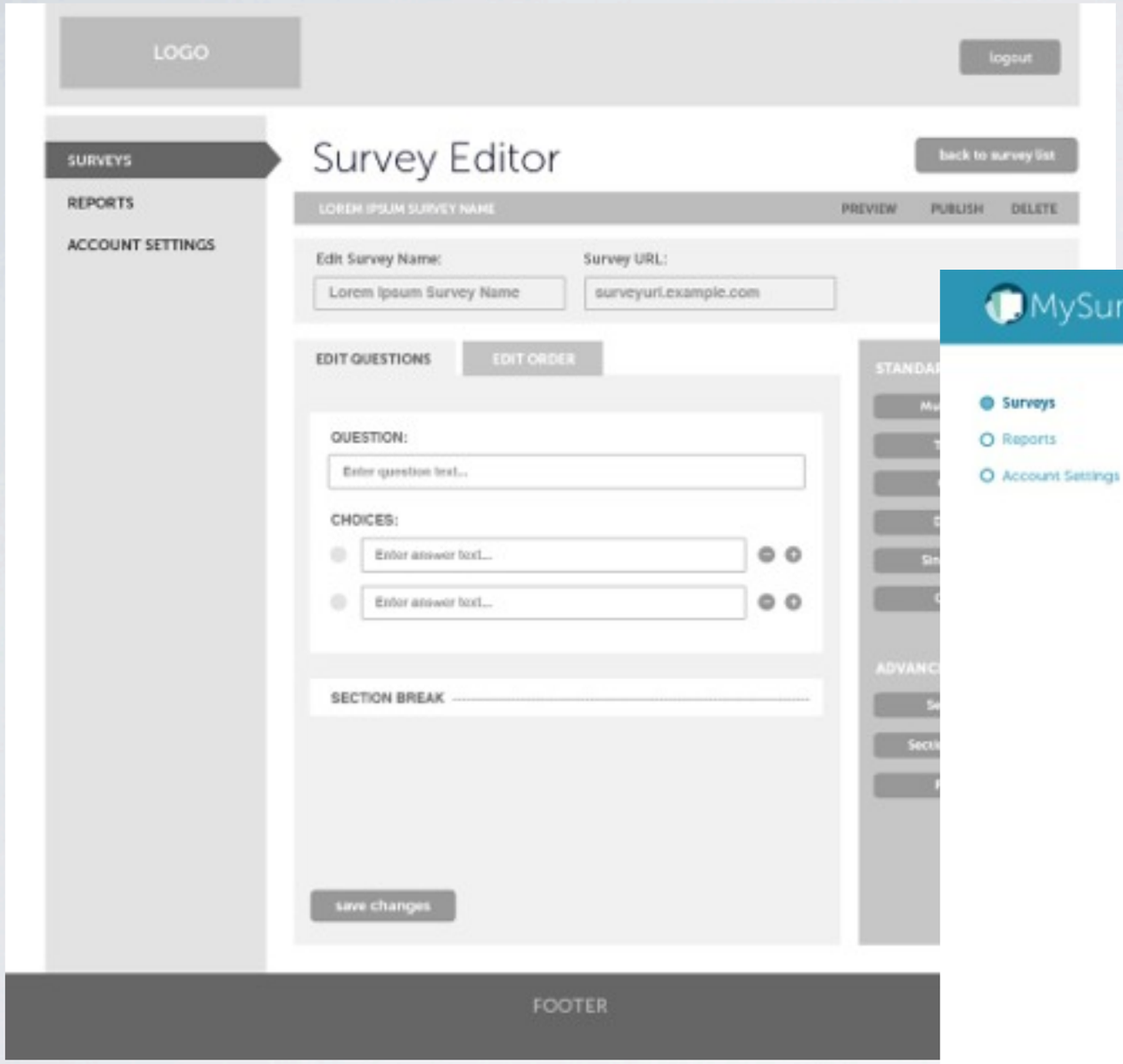


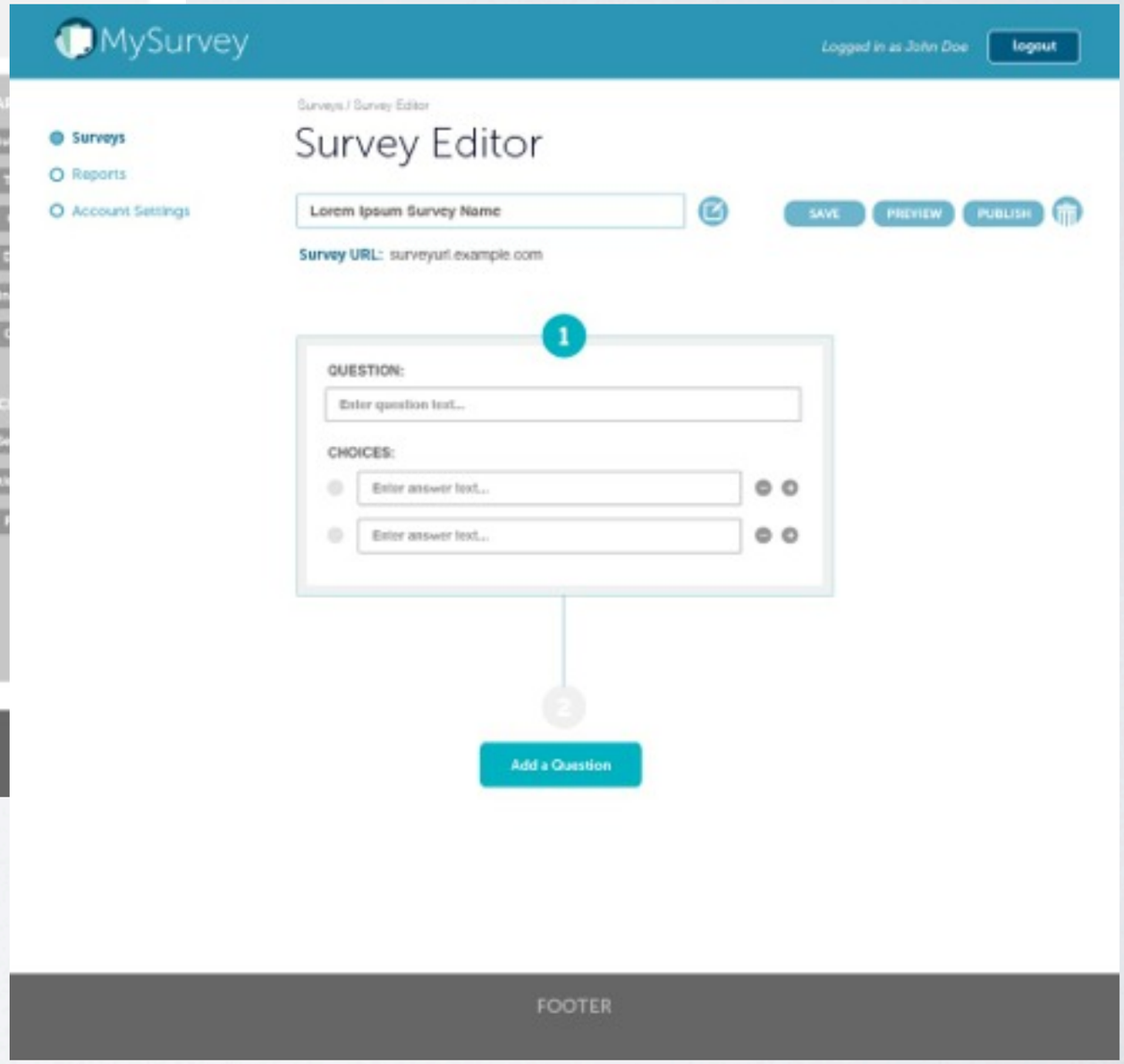
Homepage



DESIGN + UI

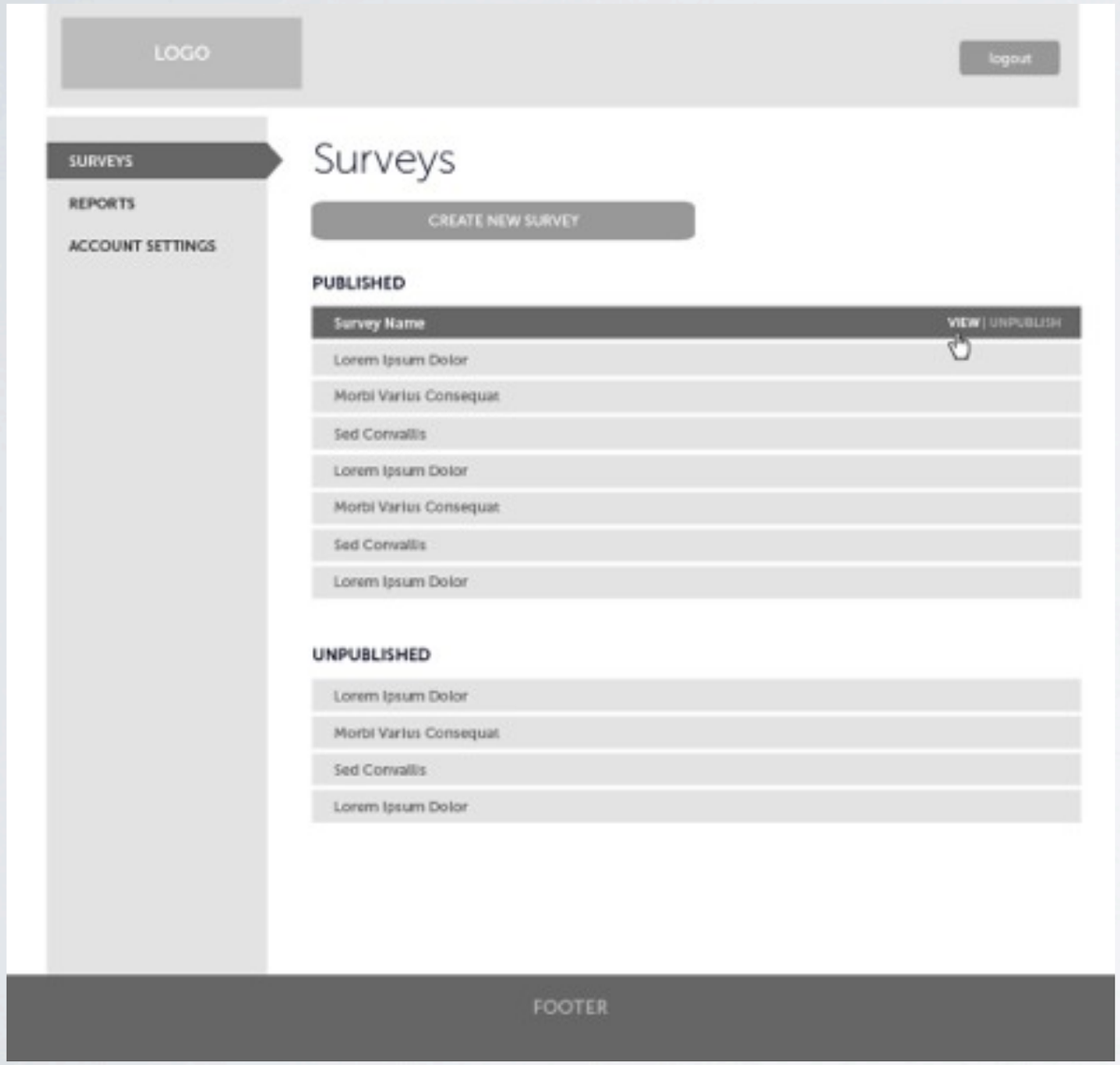
Survey Editor
(early iterations)



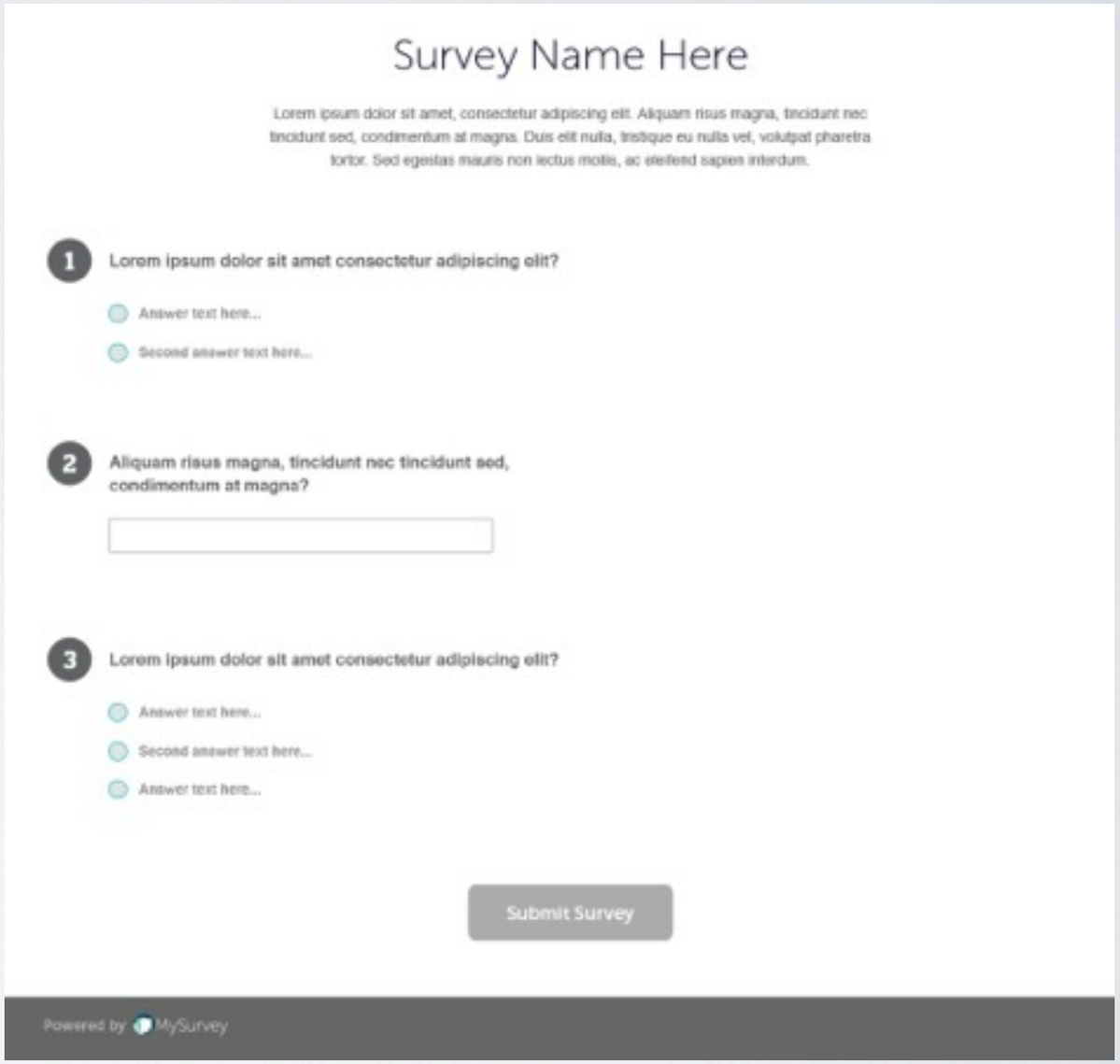


DESIGN + UI

List of Surveys



Take Survey



IMPLEMENTATION

NEW FEATURES

- More question types (Multiple select, True/False)
- Allow the survey creator to preview a survey
- Take a published survey
- Reports page with statistical charts and .csv file export
- Added “About” section and “Contact Us” page

TOOLS USED

VERSION CONTROL / BUG TRACKING

TESTING

USER STORY MANAGEMENT

DOCUMENTATION

CODING PRACTICES

Followed conventions defined by the yii framework:

- **Naming convention for code:** camel case with lowercase first letter for variables/functions, private class members with '_' prefixed, etc.
- **Directory Structure:** /themes for views. Controllers, models, tests under /protected, etc.
- **File names + url format.**
- **Database table + column names.**

Code refactoring was done at the end of each iteration

INTEGRATION PROCESS

- Every new feature was developed on a separate branch, along with branches for unit and functional testing.

- When many people worked on the same feature, individual local branches were worked on and then merged to the feature branch

- The fully tested feature branch was merged into a design branch for styling

- The tested, styled version was then merged into master and tested again

CHALLENGES

- Getting used to the yii framework took time
- Making the site user friendly by avoiding multiple separate pages
- Resolving conflicts after merges

functional
TESTING

SETTING UP AUTOMATED TESTS

Install Selenium WebDriver + PHPUnit

Write testing script using the WebDriver

Run automated tests

Re-test if UI or back-end code changes

AUTOMATED TESTING

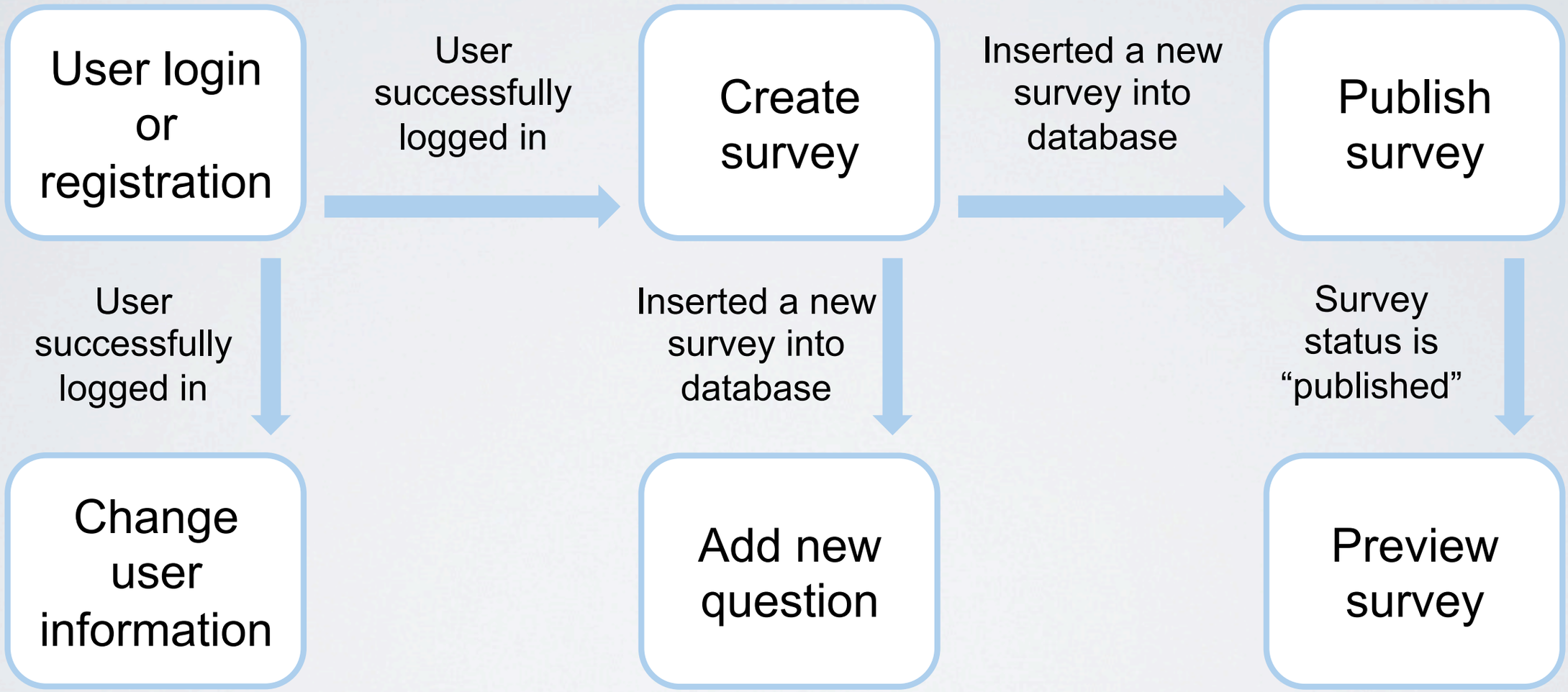
Sample functional tests:

- User can create new survey and add new question.
- User can change his/her account information.
- User can publish survey and preview survey.

Functions to be tested:

- Login, user verification.
- Database insertion / update.
- Getting survey from the database.

DEPENDENCIES



AUTOMATED FUNCTIONAL TESTING

DEMO

MANUAL VS. AUTOMATED

Manual Testing (ex: user login, registration...)

Pros:

- Low cost for test case design.
- Requires less for technical expertise.
- Tester can start testing right away.

Cons:

- High cost for execution time.
- Takes more time to complete regression testing.

MANUAL VS. **AUTOMATED**

Automated Testing (ex: create survey, update questions...)

Pros:

- Low cost for execution time.
- Allows more frequent testing.
- Reduces time to complete regression testing.

Cons:

- High cost to develop and maintain test scripts .
- Not suitable for very complex function or GUI that cannot be automated.

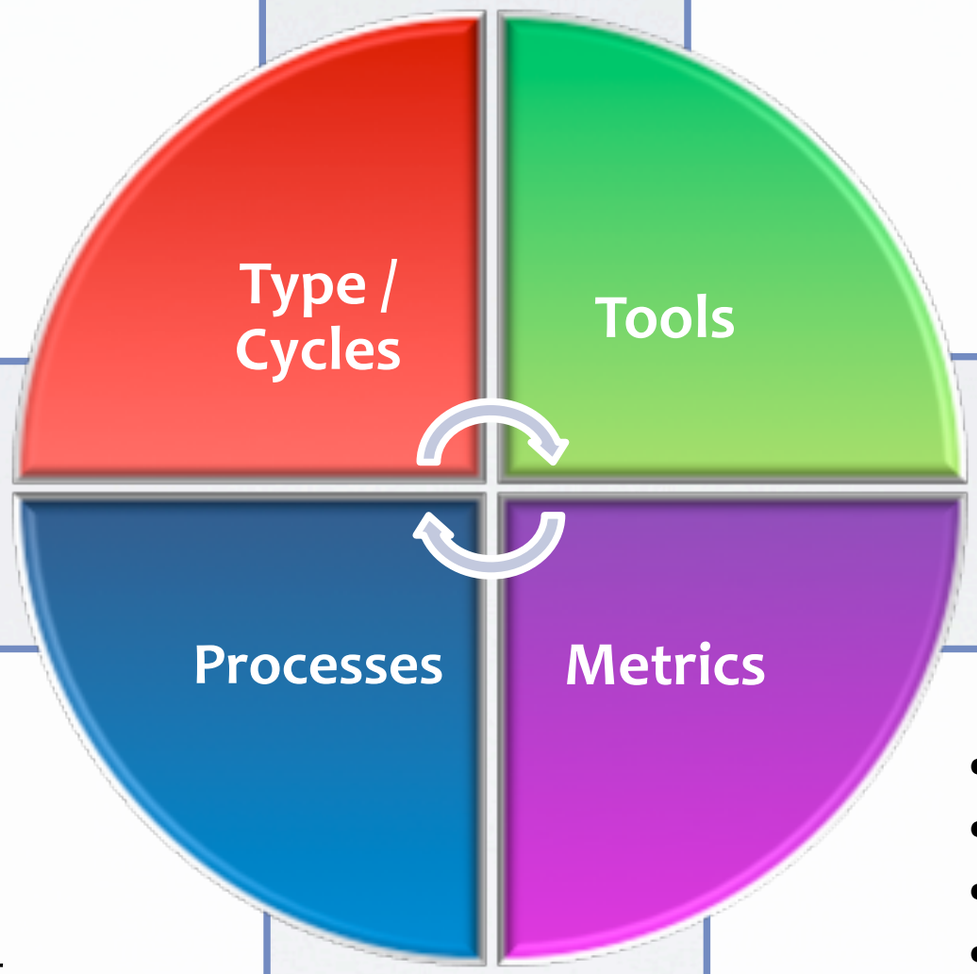
unit

TESTING

OVERVIEW

- Unit Test
- Integration Test
- System Test
- UAT

- Unit Test
  - Yii + PhpUnit
- Functional Test
  - Selenium
- CM tool: Github



- Not TDD
- Coverage: method / functionality
- Defect management
- Change management

- Defects: 27
- Resolved: 27
- Open Issues: 0
- Test Cases: 59+
- Test Rounds: 10

FEATURES TO BE TESTED

Survey Creator

- Login / Logout
- Create / Edit / Delete a survey
- Create / Edit / Delete survey questions and answers
- Preview a survey
- Publish / Un-publish survey
- View / Download / Print Report
- Report chart type change
- Account setting (password reset)

Survey Taker

- Take the survey and submit

SAMPLE TEST CASE #1

- Test Case ID: TC05002
- Test Item: Delete survey confirmation
- Test Priority: Medium
- Dependencies: TC03001
- Preconditions: Create a new survey
- Test steps: Delete
- Post condition: Server will ask the user to confirm deletion
- Actual output: no confirmation dialog
- Pass or Fail: Failed
- Defect ID: Issue #6
- Defect link: https://github.com/fydo23/mySurvey/issues/6
- Additional notes: Fixed

SAMPLE TEST CASE #2

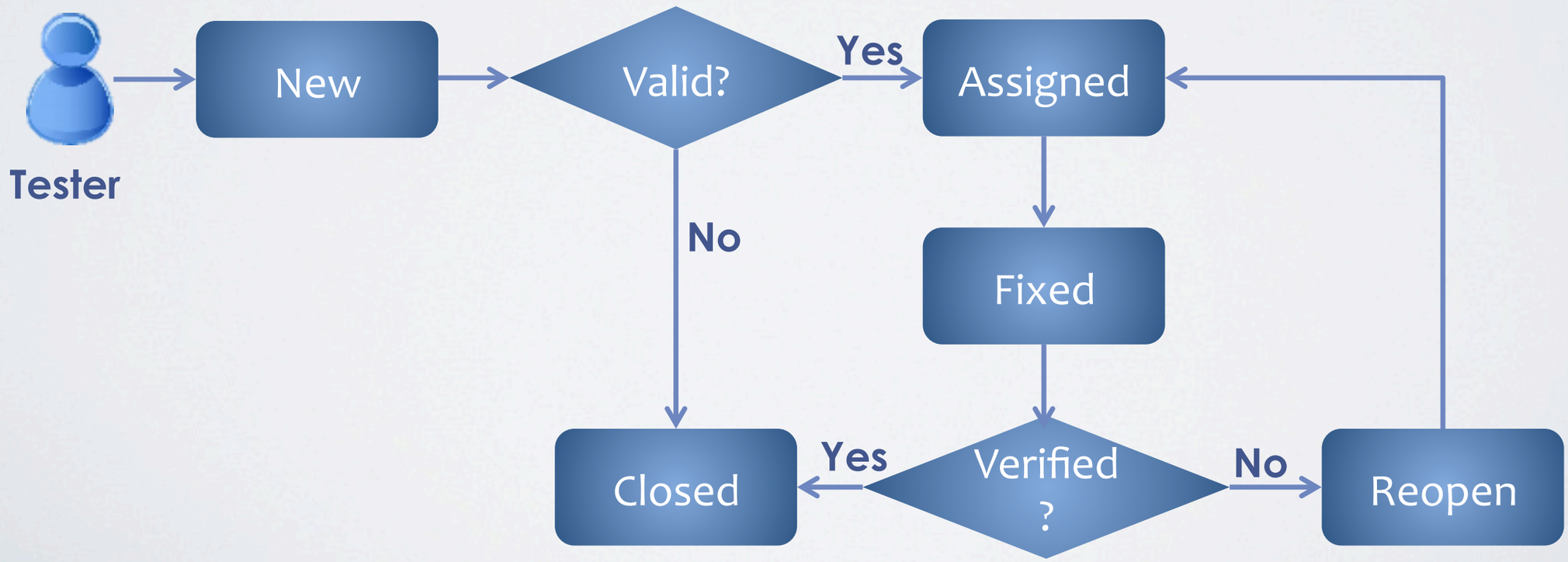
- Test Case ID: TC02003
- Test Item: Register a new user with incorrect data
- Preconditions: In MySurvey signup page
- Test steps:
  - Enter user information
  - Use invalid email address
  - Use short password (less than 8 chars)
  - Password not repeated exactly
- Input data:
  (equivalence class)
  email: (blank)
  email: aaa@bbb
  email: testmysurveywrong.gmail.com
  password: pa55!
  password: (blank)
  repeat password: (blank)
  repeat password: pa55
- Post condition: Server will ask the user to confirm deletion
- Actual output: no confirmation dialog
- Pass or Fail: Pass

DEFECT MANAGEMENT

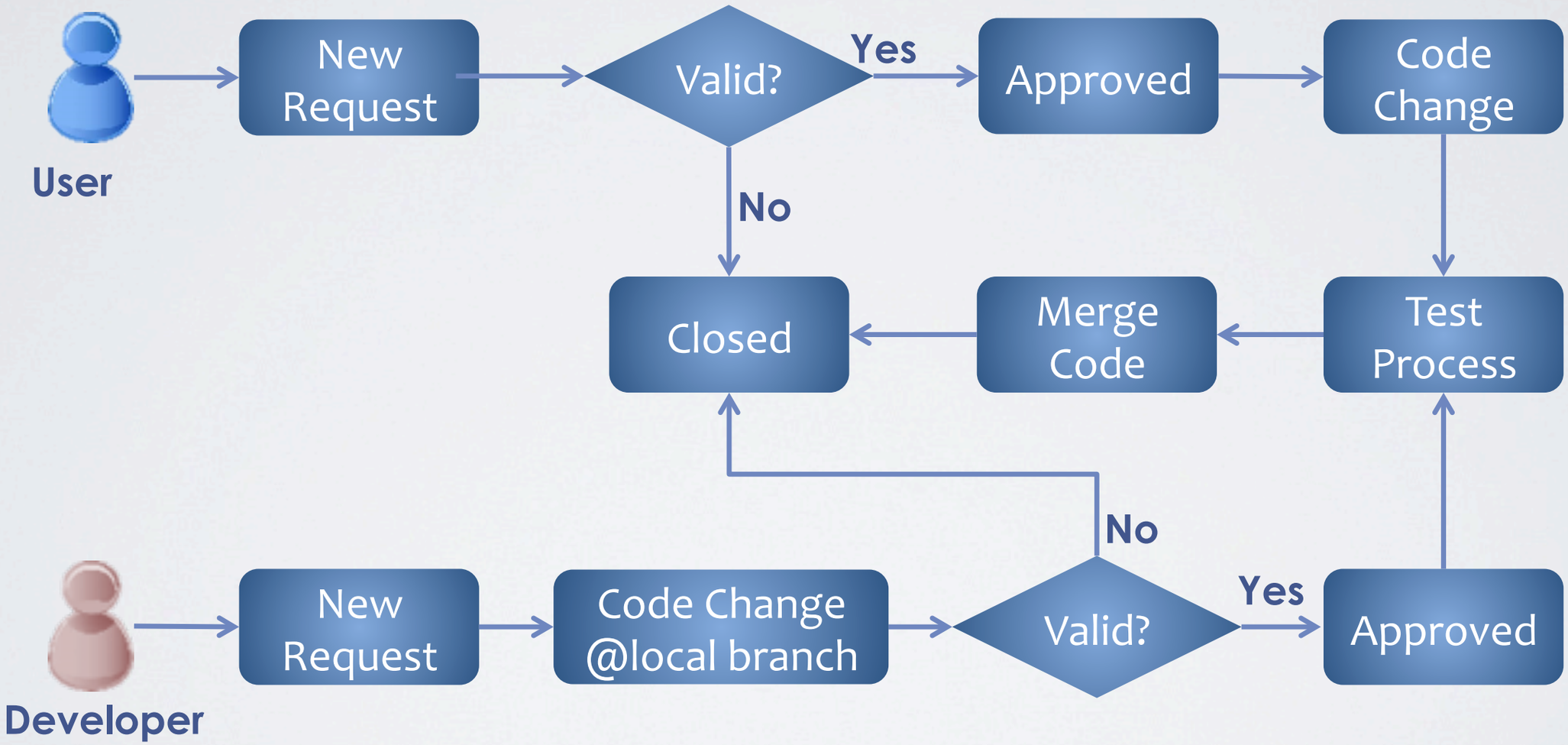
Defect tracking tools:

- Github / Mail / Team discussion

Processes:



CHANGE MANAGEMENT



Github was used as the source control / configuration management tool

Software DEMO

PROJECT *management*

METRICS

Iteration #1

- User Stories Completed: **6**
- Pivotal Tracker Velocity: **20**
- Bugs Resolved: **5**
- Open Issues: **4**
- Lines of code: **6644**

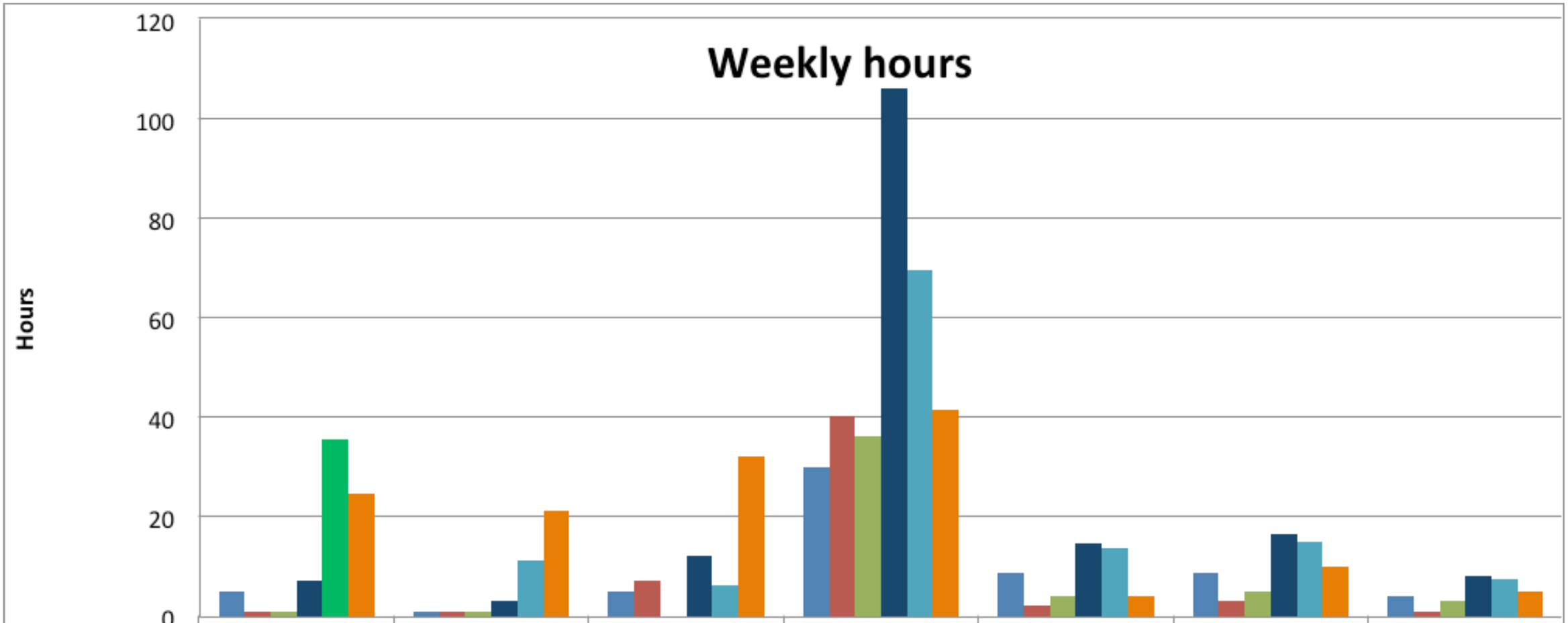
Iteration #2

- User Stories Completed: **5**
- Pivotal Tracker Velocity: **30**
- Bugs Resolved: **6**
- Open Issues: **4**
- Lines of code: **7473**

Iteration #3

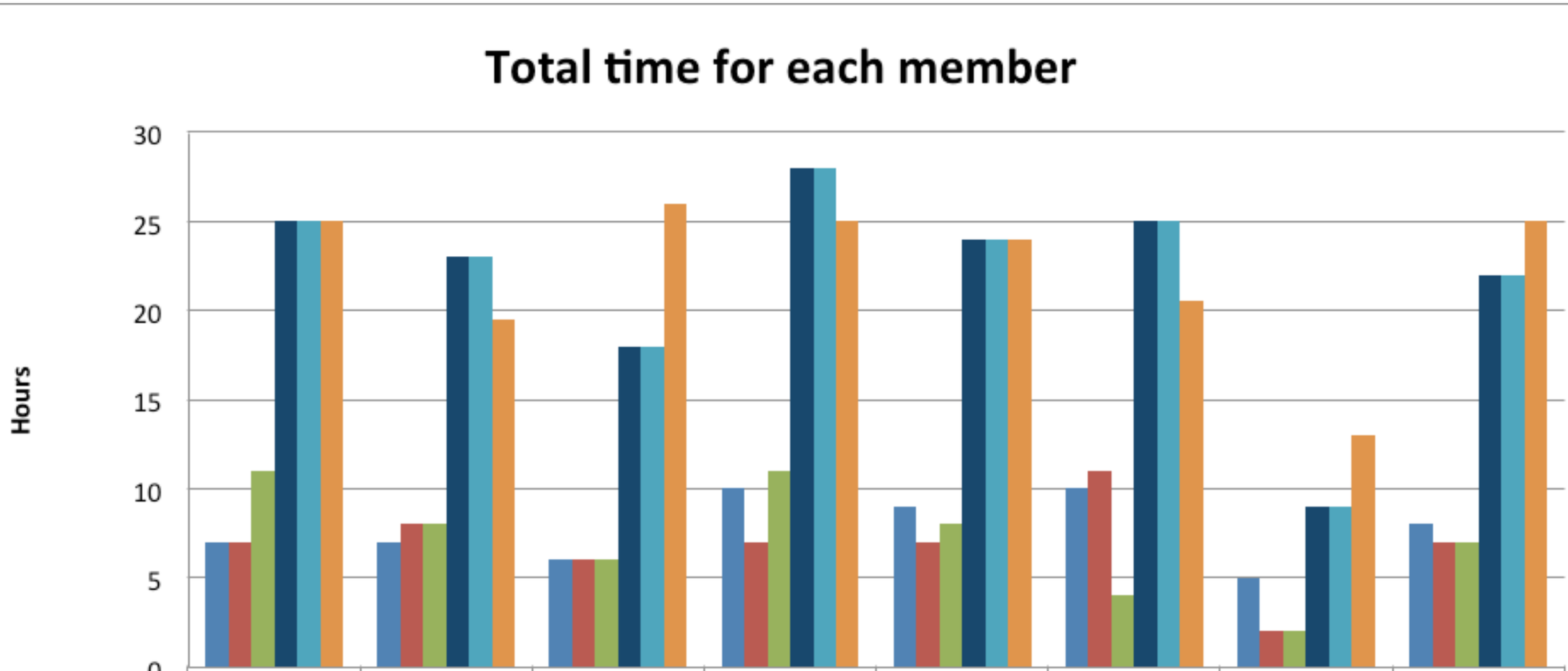
- User Stories Completed: **11**
- Pivotal Tracker Velocity: **30**
- Bugs Resolved: **17**
- Open Issues: **0**
- Lines of code: **19904**

WEEKLY HOURS



	learning	requirement analysis	design	implementation	test	communication/ management	unclassified
Week 1	5	1	5	30	8.5	8.5	4
Week 2	1	1	7	40	2	3	1
Week 3	1	1	0	36	4	5	3
Total Iter 3	7	3	12	106	14.5	16.5	8
Total Iter 2	35.5	11	6	69.5	13.5	15	7.5
Total Iter 1	24.5	21	32	41.5	4	10	5

WEEKLY HOURS



	Member 1	Member 2	Member 3	Member 4	Member 5	Member 6	Member 7	Member 8
Week 1	7	7	6	10	9	10	5	8
Week 2	7	8	6	7	7	11	2	7
Week 3	11	8	6	11	8	4	2	7
Total Iter 3	25	23	18	28	24	25	9	22
Total Iter 2	25	23	18	28	24	25	9	22
Total Iter 1	25	19.5	26	25	24	20.5	13	25

CURRENT PROGRESS

Planning Phase

- Requirements analysis
- Database Design, Wireframes & Class Diagram

Iteration #1

- User management, register, log in & log out functions (Programming & unit testing)
- Started create survey: survey title, question & answer

CURRENT PROGRESS

Iteration #2

- Completed create survey: survey title, question & answer
- Started adding different types of question to the survey

Iteration #3

- Finished adding different types of question to the survey
- Finished survey reports

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.

Plan for Support

- Plan ahead for maintenance & issues along the way.

Version Consistency

- Every team member should have the same version of the development tools.

Estimation

- Be a “Pessimist” when estimating features and iterations.
- Every team member should participate in the estimation process. We know our skills better than anyone else does.

Testing

- Start unit testing right away. Do not plan to go back & write the tests

thank
YOU

MySurvey