Innovation Drive and Technology Expo

Test Plan Documentation

Chris Ciolek

Nicholas Spencer

Maxx Achtman

Apolonio Cazares

[**Introduction**](#_shljhw9jjp4y) **4**

[Goals and Objectives](#_1u0dop8x61rw) 4

[Statement of Scope](#_1u0dop8x61rw) 4

[Functional Requirements](#_qftmc61dnx8f) 4

[Non-Functional Requirements](#_dcwfq2ewx6bg) 5

[Hardware Requirements](#_nlbsszu6ahxm) 5

[Software Requirements](#_zyer5f5w2l) 5

[Inverse Requirements](#_xbxx9yt09dc) 5

[Deliverables:](#_hc61j4wrzehq) 6

[Inputs](#_uo4q8gjgoh9s) 6

[Processing](#_fou09ez8y8qk) 6

[Output](#_pupz23ctd7t5) 6

[Major Constraints](#_1u0dop8x61rw) 6

[**Test Plan**](#_k946br9gspb) **6**

[Software (SCIs) to be tested](#_mqqckita2v93) 6

[Interfaces](#_lj32hrxha7uj) 6

[Homepage](#_9j0km25cu8gq) 6

[Registration Page](#_ac4kvx61f25c) 7

[Digital Registration Waiver](#_wu0cofft3w8h) 7

[Registration for Supplier or Ford Presenter](#_r5ddyq66qd0m) 7

[Registration for Evaluators](#_acbd4x13orge) 7

[Digital Waiver for Technology Submission](#_qmy3t3iiuwct) 7

[Technology Submission Form](#_d0weclape0wv) 7

[Information Tab](#_tnv1hhv1q6wy) 7

[Event Information](#_y74z39nw9aib) 8

[Event Map](#_wgzp1mtgeg9q) 8

[Event Layout Map](#_kvavqefgxz9w) 8

[Contact IDTE](#_1t0r0gu3cug2) 8

[Admin Page](#_a5zpbdufygml) 8

[Edit Technologies](#_wqq9eeytm1bm) 8

[Edit Event Dates](#_bgirvt5fgpjr) 8

[Open / Close Registration](#_aizdpa8jyd75) 8

[Attendee Checkin](#_qwviib2t4s7s) 9

[QR Sign In](#_od2bm93k1mjt) 9

[Edit Attendee](#_irzcuaeiwuzb) 9

[Testing Strategy](#_mqqckita2v93) 9

[Unit Testing](#_5rvl7cqirj71) 9

[Integration Testing](#_5rvl7cqirj71) 9

[Validation Testing](#_5rvl7cqirj71) 10

[High-order Testing](#_5rvl7cqirj71) 10

[Recovery Testing](#_hqvwx1b4rn2a) 10

[Security Testing](#_pyiaq2y3xp12) 10

[Stress Testing](#_9yyg4aygk3nr) 10

[Performance Testing](#_n3axw1iybfe) 11

[Alpha/Beta Testing](#_brz5nnnpgz9l) 11

[Pass / Fail Criterion for all Validation Tests](#_nwbb1j2p69zo) 11

[Testing Resources and Staffing](#_bd2bc5nmgmvp) 11

[Test Record Keeping](#_bd2bc5nmgmvp) 11

[Test Schedule](#_bd2bc5nmgmvp) 11

[**Test Procedure**](#_yco3ij6xfhqg) **12**

[Software (SCIs) to be tested](#_om14rp63yfd8) 12

[Testing Procedure](#_om14rp63yfd8) 12

[Unit test cases](#_aqp8xfoc1ki2) 12

[Homepage](#_riknwudk2sc) 12

[Registration Page](#_fcsi1z3d5bf3) 12

[Digital Registration Waiver](#_1eezky4yrskb) 12

[Registration for Supplier or Ford Presenter](#_ie9fgiy8wnuu) 13

[Registration for Evaluators](#_60kf5qlbb2cn) 13

[Digital Waiver for Technology Submission](#_y2i861ucqis8) 13

[Technology Submission Form](#_1t888jxmq53m) 13

[Information Tab](#_bs6ef6dighy7) 13

[Event Information](#_t5umusvruved) 13

[Event Map](#_e72acc7iqruf) 13

[Event Layout Map](#_o3ua171krkgz) 14

[Contact IDTE](#_u9n50uaem59y) 14

[Admin Page](#_dz7j5u5menxt) 14

[Edit Technologies](#_s8n8f6tdynm2) 14

[Edit Event Dates](#_nocuvy8wfg6v) 14

[Open / Close Registration](#_hl8uzel8x0ea) 14

[Attendee Checkin](#_8d9zb2t8qih3) 14

[QR Sign In](#_r6x475vs82ly) 15

[Edit Attendee](#_v8m4672c12vl) 15

[Integration Testing](#_cmb6bzfikftx) 15

[Validation Testing](#_fsgppbpaujfo) 15

[High-order testing](#_yffp2h8ctkb) 16

[Recovery Testing](#_xp32uaf4l1qe) 16

[Security Testing](#_xp32uaf4l1qe) 16

[Stress Testing](#_xp32uaf4l1qe) 16

[Performance Testing](#_xp32uaf4l1qe) 17

[Alpha / Beta Testing](#_xp32uaf4l1qe) 17

[Pass / Fail Criterion for all Validation Tests](#_xp32uaf4l1qe) 17

[Testing Resources and Staffing](#_33ps0ws8cik) 18

[Test Record Keeping and Test Log](#_33ps0ws8cik) 18

[**Test Cases**](#_u1tobedulc0h) **19**

[**Project Management**](#_ddiywbqhoeav) **19**

[Tracking and Control Mechanisms](#_go4kylh4v8wn) 19

[Quality Assurance and Control](#_g7l1sxwqmxua) 19

[Change Management and Control](#_ueifuuhbejyd) 19

[Tools Used](#_8pr71eh7l3lr) 19

[Project Schedule](#_5ejbmlnzcxw4) 19

[Project Task Set](#_ko35crlpxyy2) 19

[Functional Decomposition](#_ko35crlpxyy2) 21

[Task Network Diagram](#_ko35crlpxyy2) 22

[Timeline Chart](#_ko35crlpxyy2) 23

[Project Schedule](#_ko35crlpxyy2) 23

# Introduction

## Goals and Objectives

The goal for this software is to allow users to select dates and register to be a participant in Fords IDTE event, and receive confirmation emails regarding the event. Additionally, each user will have a generated QR code on their ticket that will let them check in when they arrive at the IDTE.

## Statement of Scope

### Functional Requirements

* + - 1. User can register as a supplier, presenter, or evaluator
      2. A waiver needs to be signed before a user can sign up to attend the event
      3. Attendees for the event can modify their registration at any point before an event
      4. Administration confirms a registration from a requested attendee
      5. Administration can view how many attendees there would be on a given day for the event
      6. A superadmin would be created that can have access to administrative accounts
      7. Administration can change dates of events
      8. Administration can open or close the registration dates for events
      9. Email that is sent to the attendee contains a QR code with their information provided during registration
      10. Badges would be printed on the day of the event by administration

### Non-Functional Requirements

* + - 1. Application can run on mobile
      2. Application can run on any web browser
      3. Database is secure and cannot be accessed outside of select individuals
      4. User information for attendees can only be viewed by their own person
      5. Information stored in database is only for IDTE

### Hardware Requirements

* + - 1. Windows OS
      2. Mac OS
      3. Web camera
      4. Mobile Phone
      5. Printer

### Software Requirements

* + - 1. Node.Js
      2. HTML
      3. CSS
      4. JavaScript
      5. Java
      6. Sqlite

### Inverse Requirements

N/A

### Deliverables:

#### Inputs

* + - * 1. User credentials
        2. First Name
        3. Last Name
        4. Email Address
        5. Association with event
        6. Dates to attend event

#### Processing

* + - * 1. Data from inputs to database
        2. Generate QR code

#### Output

* + - * 1. Email sent to user with QR code
        2. Badge printed at event

## Major Constraints

Lack of connectivity to internal Ford Servers for proper testing. Can only run live testing locally until access is granted. Without it, we would be sending the customer a non-fully tested product.

# Test Plan

We want the product to be bug free. In order to reach this conclusion, we would have to confirm different portions of our project. We have to make sure the databases we have constructed talk to the UI accordingly. We need to make sure that users can sign up for events, submit a technology, and be able to edit any credentials. Along with this, it is also crucial to make sure that email functionality works as expected. Our flow states that when a registration is submitted, an email is sent with a QR code. The QR code needs to work with a scanner on the day of the event for a badge print out. Doing all of this will ensure the customer would receive a fully functional and bug free product.

## Software (SCIs) to be tested

### Interfaces

#### Homepage

We will be accessing the homepage link to make sure that whenever the user first accesses our site, they will be directed to the homepage

#### Registration Page

Our registration page would allow the user to select from a Supplier, Ford Presenter, or Evaluator through a UI button.

#### Digital Registration Waiver

A digital waiver would be presented to the user to confirm they abide to the specifications for registration. This would involve a checkbox to act as a digital signature along with a submit button.

#### Registration for Supplier or Ford Presenter

Following a digital signature, a user would be shown with multiple text fields to enter for a completed registration. Depending on selecting a Ford Presenter or Supplier would have different fields to enter. Once the submit button is selected, they will be sent an email with a QR code to scan at the event.

#### Registration for Evaluators

Following a digital signature, a user would be shown with multiple text fields to enter for a completed registration. As an Evaluator, certain values will be ignored when compared to a Supplier or Ford Presenter as per the customer request. Once the submit button is selected, they will be sent an email with a QR code to scan at the event.

#### Digital Waiver for Technology Submission

By clicking the Technology Submission selection on our navigation bar, the user would be directed to submit to a Technology Submission Digital Waiver. This behavior would be the same as registering to attend the event

#### Technology Submission Form

After a digital waiver is signed, the user would be able to enter in credentials that are necessary for a Technology Submission. Once the user selects the Submit button, they would be sent an email with a confirmation of the submission.

#### Information Tab

Hovering over the Information tab would allow the user to view options such as Event Information, Event Map, Event Layout Map, and Contact IDTE.

#### Event Information

When a user selects this link, they would be able to view information about our event. **This will be updated once we get the notes from our client with the information we would need.**

#### Event Map

When a user selects this link, they would be shown a viewable map of how to get to the parking lot for the event. **This will be updated once we get the information from our client.**

#### Event Layout Map

When a user selects this link, they would be shown a map of the event with tent locations. **This will be updated once we get the information from our client**.

#### Contact IDTE

When a user selects this link, they would be shown contact information for those in charge of IDTE. **This will be updated once we get the information from our client.**

#### Admin Page

When an admin selects this link, they would be shown the Admin page. This is a view only accessible by admins of the event. Normal users cannot view this page.

#### Edit Technologies

When this button is selected, the user is shown a view of all technologies that are currently in the database. In this view, the user can edit, delete, or add new technologies.

#### Edit Event Dates

When this button is selected, the user would be able to edit the dates of the event. Any updates made here would be recognized by the website. Only an admin can use this feature.

#### Open / Close Registration

This view would allow an admin to make registration for a technology or attendee be accessible by a normal user. Once the time value is reached, then a normal user cannot register and would have to request a registration from an admin.

#### Attendee Checkin

When a user would select this button, they would be taken to an option to register an attendee. This could be either through a QR code scan or attendee lookup.

#### QR Sign In

On the days of the event, a user would take their QR code and scan it. In order for that to be accessed, those at the front desk would have to select this option. Once a scan is successful, a badge would be printed with a connected printer.

#### Edit Attendee

When this button is selected, the user is shown a view of all technologies that are currently in the database. In this view, the user can edit, delete, or add new technologies.

## Testing Strategy

### Unit Testing

When we conduct our unit testing, each use case that we have implemented would be tested. The testing will be conducted through white box where each use case prepared would be tested individually. The use cases would be tested by passing through custom made data and see if any errors occur. Along with features mentioned above, entry and exit conditions would also be met. Any Unit Testing would be conducted by the programmer who created the use case. If needed, we would have a second programmer be a tester as well to see if anything needs to be updated or changed prior to a final release.

### Integration Testing

**This testing can only be conducted once we have access to Ford Servers. If that is not possible, then this portion of testing cannot be conducted. This has already been mentioned in our major constraints in section 1.3.**  If the above is met, we can begin testing on our client’s servers and measure our results. Various testing scenarios that we would do can be denoted by the following. Conduct a full workflow of registrations as if we were a normal user. Verify admin rights work as expected on a live server. Once all of the conditions are met, we will feel confident with launching our software to their servers.

### Validation Testing

This testing that we conduct would be run with the full understanding of being a normal user using our software. This is important to keep in mind since as the programmers and testers, we know HOW this product would be built and run as expected. This approach can be used with black box testing where we test all components of the product at once. Any issues that could be found here can be denoted accordingly in various test cases. The goal of our validation testing is to take a fully functional, bug free product to the customer so they can walk away happy and know that this is something that is usable for them.

### High-order Testing

The below test cases would be used to cover for different conditions that we would not be putting a focus on during other testing scenarios. These testing scenarios would be run when we reach big milestones in our project.

#### Recovery Testing

This testing would be conducted to confirm that if a database were to lose connection to our site, the information that is stored would not be lost. The biggest idea here is to cover for loss of technologies or users.

#### Security Testing

We want to use this testing to be sure that the site is secure. With having people’s personal information on our site, we want to be sure that unless you are an admin, only your own person can view your own personal information.

#### Stress Testing

We want to use this testing to see how our site would run when there is an overload of processes being run on a computer. This can help us see if any changes are needed to be made to our site to prevent any crashes from happening.

#### Performance Testing

We want to use this kind of testing to make sure that our product is running to the best of the ability. Since we have different kinds of users using our site, we have to make sure that speed is consistent regardless of bandwidth from a router.

#### Alpha/Beta Testing

We want to run testing levels when we send updates to the client. Since this product is in use with a team of people, we want to be sure that all parts of the project works. The client would want to have Technology Concepts completed by a certain date, so testing will be put on a bigger emphasis in said phases.

#### Pass / Fail Criterion for all Validation Tests

We are defining clear specifications for when a test case passes or fails. A test case is considered passing when all criteria are met. If there is any criteria that does not match, then the test case would be considered as failed and be submitted as a defect in the backlog.

## Testing Resources and Staffing

We will be using various resources to carry out testing our software. The below would be the testing resources

* Google Sheets to store test cases
* Personal PC’s for on the fly testing
* Connecting to customer PC with on server connection
* Trello for task tracking and defect logging

## Test Record Keeping

We are covering all Test Case keeping and products in section 3.5 of our documentation. For information regarding these topics, please see the above mentioned section.

## Test Schedule

The below plan is when testing will be conducted during each phase

**Project Test Plan**

3/1/2020 - 3/16/2020

**System Testing**

3/23/2020-3/30/2020

**Generate Testing Report**

4/1/2020 - 4/16/2020

# Test Procedure

## Software (SCIs) to be tested

Please refer to listing 2.1 to view the Softwares that would be tested. This contains the full list of all things that would be prepared and tested prior to a final release to the customer.

## Testing Procedure

### Unit test cases

This testing will be conducted by running the smallest unit of software. We will be testing the important paths to find any errors in our code. This will be conducted through a white box testing method.

#### Homepage

We will make sure when a user enters our website through the provided URL, they will be shown our homepage. This can be conducted using various web browsers and see how the page looks like on a mobile device through adjusting the screen size on the computer.

#### Registration Page

We will make sure when a user selects the Registration tab on the Navigation Bar that they will be directed to the Registration Page. This can be conducted using various web browsers and see what the page looks like on a mobile device.

#### Digital Registration Waiver

We will make sure that the features of a checkbox are working as expected. When a user selects the checkbox, then we can define that the user has agreed to the conditions met. The button we are using would also be disabled / enabled depending on the checkbox selection.

#### Registration for Supplier or Ford Presenter

We will make sure that the user can enter text values in all fields and confirm it is added to the database and sent with an email. We will use multiple user names for this step to make sure the flow works as expected.

#### Registration for Evaluators

We will make sure that the user can enter text values in all fields and confirm it is added to the database and sent with an email. We will use multiple user names for this step to make sure the flow works as expected.

#### Digital Waiver for Technology Submission

We will make sure that the features of a checkbox are working as expected. When a user selects the checkbox, then we can define that the user has agreed to the conditions met. The button we are using would also be disabled / enabled depending on the checkbox selection.

#### Technology Submission Form

We will make sure that the user can enter text values in all fields and confirm it is added to the database and sent with an email. We will use multiple technology names for this step to make sure the flow works as expected.

#### Information Tab

We will make sure that a drop down is generated for the user to make a selection. This can be conducted when hovering over the Information Tab or by selecting the information tab when on a mobile device.

#### Event Information

We will make sure when the user selects the Event Information option from the dropdown, they will be shown the information that is provided by the customer.

#### Event Map

We will make sure when the user selects the Event Map option from the dropdown, they will be shown the information that is provided by the customer.

#### Event Layout Map

We will make sure when the user selects the Event Layout Map option from the dropdown, they will be shown the information that is provided by the customer.

#### Contact IDTE

We will make sure when the user selects the Contact IDTE option from the dropdown, they would be shown the information that is provided by the customer.

#### Admin Page

This will cover two special scenarios. The first scenario would make sure that a normal user cannot access this page. The second scenario would be when an admin of the site accesses this page, then they will be taken to the new view.

#### Edit Technologies

We will make sure when an admin were to make this selection, they would be shown a list of all technologies that are currently submitted. Here, an admin can add, delete, or edit current submissions and updates would be run on the database. Behind the scenes, we would make sure that the correct values are being passed in to the right pointers.

#### Edit Event Dates

We will make sure a change in event dates is possible by an admin. Once the date value is changed, we can verify this by looking at values that are passed into the database.

#### Open / Close Registration

We will make sure an event occurs where registration can be open or closed that is not controlled by dates. This is an admin controlled feature and can be controlled by use of a drop down and submission button.

#### Attendee Checkin

We will make sure an attendee can be checked into the event from lookup of name here. This will talk to our database to make sure the attendee exists.

#### QR Sign In

We will make sure when a QR code is scanned, it will be able to talk to the database and find the Attendee ID that is automatically generated upon registration and pull it from the database.

#### Edit Attendee

We will make sure when an admin were to make this selection, they would be shown a list of all technologies that are currently submitted. Here, an admin can add, delete, or edit current submissions and updates would be run on the database. Behind the scenes, we would make sure that the correct values are being passed in to the right pointers.

### Integration Testing

**This testing can only be conducted once we have access to Ford Servers. If that is not possible, then this portion of testing cannot be conducted. This has already been mentioned in our major constraints in section 1.3.**  If the above is met, we can begin testing on our client’s servers and measure our results. Testing scenarios that could be conducted are run through various scenarios.

For one, we would be running tests on every part of the program as individual test cases. This would include testing strictly Technology Registration, Attendee Registration, Database editing, etc. until all components are tested and return successful. The goal of each sub test is to run timing and see how the behavior acts. If there are any problems between our software and the client’s servers, we would want to take care of these issues prior to the final release of the customer.

Another testing scenario we will conduct is running the website from our client’s laptop and allowing them to take control. This testing can show us how a computer controlled through the client’s IT service behaves on our site. If there are any discrepancies, they will be noted and adjusted prior to a final release.

### Validation Testing

This testing will be conducted to validate our entire software. This means we will be running through all components of the website as if we were a normal user. Testing scenarios would be used through Black Box testing methods. We will have one person on our team designated to conduct the validation testing for our website. All testing scenarios that would be conducted can be seen in section 3.2 to make sure we look at all components of our website. Along with functional testing, we would also be running User Interface testing to make sure items such as buttons, text fields, labels, etc. all match and are consistent throughout the website. If there are any errors that come up during the testing, they would be documented and flagged with a defect severity level of 1-4. One being a crash and 4 being a cosmetic defect. Our goal is to release a product that has as few bugs as possible. The only bugs that we would consider to be passable (with a future update in mind) would be level 3 and level 4 defects.

### High-order testing

#### Recovery Testing

The goal of this kind of testing is to make sure no data would be lost during any database changes. This main focus would be talking between the backend database with the front end view that the user could see.

We want to make sure that during the transition period between a user change and a database update, any cut of communication between the link would not affect the database.

#### Security Testing

Since we are handling personal information of users and technologies that would be considered confidential, security is a big factor.

Our security testing would be run to make sure that only users who submit a technology or attendee registration can only see their personal information. The best practice for this is to make sure the link we send to the user’s email that they will direct them to their own information.

The only users who would be able to view all values in our database are admins. Admins are those who are part of IDTE and have valid credentials.

#### Stress Testing

This test method focuses on the ability to run concurrent transactions at the same time. This testing will be run by having multiple programs opened at the same time and seeing how our website can handle the memory usage. Our goal is to make sure that our website does not crash when multiple programs are running.

Another test that will be conducted is seeing how the website reacts on a network with multiple tabs opened. The goal of this testing is to make sure that during the network amount being used, the website would never lose connection. Timings may be increased, but that will be denoted in testing results.

#### Performance Testing

We are testing the software to make sure it meets the criteria for the clients requirements. All of this testing will be conducted on site and run with the hardware that our website will be used for. The performance bounds that we want to focus on would be the following:

* Response time when running a database search
  + Best case would be updated immediately
  + Worst case would be 2 seconds
* Response time when moving between pages
  + Best case would be updated immediately
  + Worst case would be 2 seconds
* Response time with receiving an email after registration
  + Best case would be updated immediately
  + Worst case would be 2 seconds

#### Alpha / Beta Testing

For Alpha Testing, our main goal for a final submission is having Technology Registration completed. Prior to giving the client access to conduct this part of the website, we want to make sure the workflow works as defined by our client.

For Beta Testing, our main goal for a final submission is having Event Registration completed. Prior to giving the client access to conduct this part of the website, we want to make sure the workflow works as defined by our client.

#### Pass / Fail Criterion for all Validation Tests

Testing that will be conducted can be seen in Section 3.5 with how we handle record keeping. The criteria for a test case passing or failing would fall underneath if our expected result matches our actual result. If the test case would fail, then a defect would be written and the programmer would be notified.

Multiple test runs will be conducted throughout the process of the project to make sure that existing functionality is not broken when new functionality is introduced.

## Testing Resources and Staffing

The resources we have available to us would help us make sure that the entire software is tested correctly. The following would be used to meet that goal.

* Google Sheets to store test cases

Using Google sheets would allow us to organize the kinds of test cases that would be conducted, when it is being run, how many test runs are conducted from beginning to end, and any defects that may exist in a given test run. This will show improvements throughout the software life cycle.

* Personal PC’s for on the fly testing

Testing conducted on our personal PC’s would allow us to test new code. We could also conduct this through black box and white box testing since we can launch our code to a local server and run it through a local site connection. Any errors that occur here can be fixed and adjusted before conducting a test run.

* Connecting to customer PC with on server connection

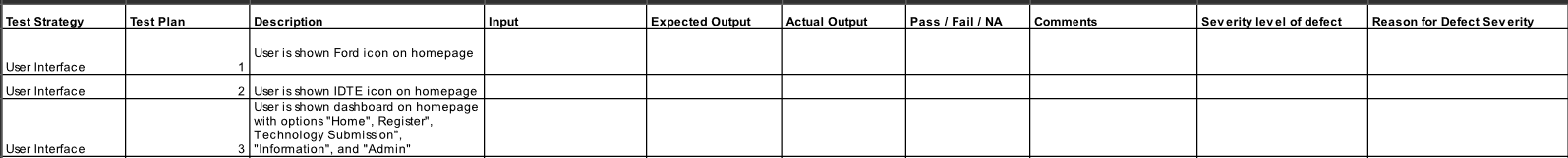
Testing conducted on a customer PC would allow us to see the results running within the confines of what we are given. Testing that is run here can have us understand what the website would be like in a live setting. This will force us to run strictly in black box testing and may return in different results than previously anticipated.

* Trello for task tracking and defect logging

If a defect exists during a test run. We will use Trello to store the defect. Along with the type of defect, we would provide necessary steps so the programmer would know how to fix the bug. Once the defect is fixed, a tester would conduct the finalization of the software and close the task.

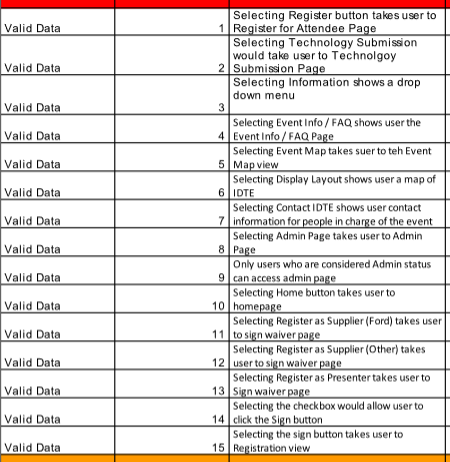
## Test Record Keeping and Test Log

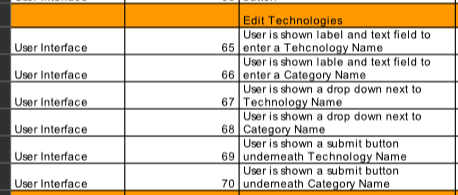
We will be using a Google Sheet to conduct all testing needed for the project. Tabs are created to specify the test strategy. Test strategies would be denoted as the following: Valid Data, Invalid Data, User Interface, and Exporting Data. All of these test cases would then be placed into a test run where we would provide the listings as shown in the following screenshot. If any test cases fail, they will be denoted in a separate task that will be used to add to our backlog on Trello. One final test run will be run before we hand over the project to the customer.

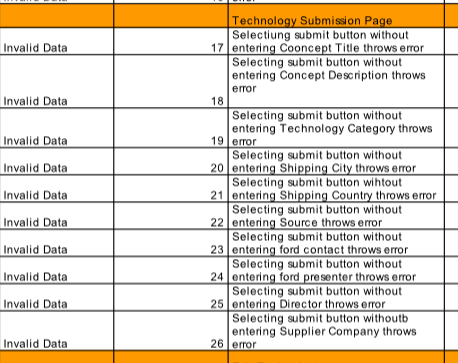


# Test Cases

Below is a sample view of some of our test cases. Our actual test cases can be found using the link at the bottom of this section.







Link to test cases: [CLICK HERE](https://drive.google.com/file/d/1XPkBR5359MXlg5lZw-4pubM4i45MODKN/view?usp=sharing)

# Project Management

## Tracking and Control Mechanisms

### Quality Assurance and Control

* All classes will have test stub evaluations before implementation
* Each group member will carry out testing on both their own and others’ work
* A designated group member will conduct final testing

### Change Management and Control

* Project artifacts will be tracked and maintained using a GitHub repository
* Proposed changes should be submitted with a brief but descriptive summary of what was changed, i.e. added, removed, or modified and explain why
* Changes to artifacts will be announced through the discord server and the group may vote to revert the changes

### Tools Used

* GitHub
* Discord
* Visual Studio
* Google Drive
* Enterprise Architect

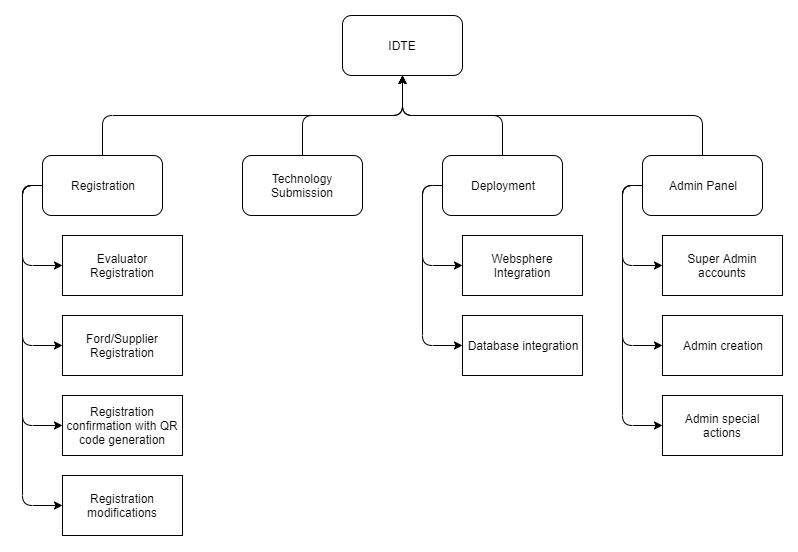
## Project Schedule

### Project Task Set

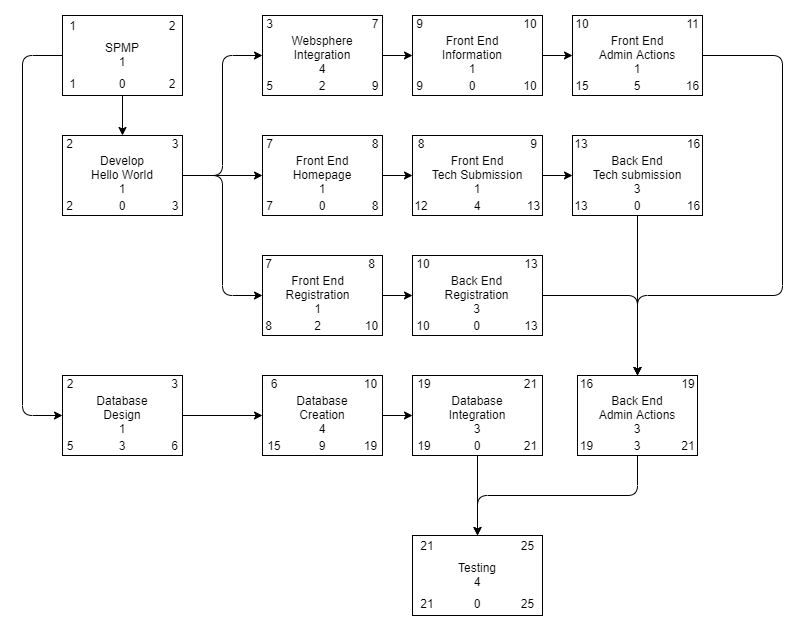
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Task | Time  (Weeks) | Predecessor Tasks | Successor Tasks | Start Date (ES) | End Date (EF) | Slack Time | % Complete |
| SPMP | 1 |  | Develop Hello world | 1 | 2 | 0 | %100 |
| Develop Hello World | 1 | SPMP | Websphere Node Integration | 2 | 3 | 0 | %100 |
| Websphere Node.js integration | 4 | Hello world | Testing | 3 | 7 | 2 | %100 |
| Front End Home page | 1 | Hello world | Front End Tech Submission | 7 | 8 | 0 | %100 |
| Front End - Registration | 1 | Hello world | Back End - Registration | 7 | 8 | 2 | %75 |
| Front End - Technology submission | 1 | Hello world | Back End - Technology submission | 8 | 9 | 4 | %75 |
| Front End - Information | 1 | Websphere  Integration | Testing | 9 | 10 | 0 | %50 |
| Front End - Admin Actions | 1 | Hello world | Front End - Admin Actions | 10 | 11 | 5 | %25 |
| Back End - Registration | 3 | Front End - Registration | Testing | 10 | 13 | 0 | %100 |
| Back End - Technology submission | 3 | Front End - Technology submission | Testing | 13 | 16 | 0 | %100 |
| Back End - Admin Actions | 3 | Front End - Admin Actions | Testing | 16 | 19 | 3 | %100 |
| Database Design | 1 | SPMP | Database Creation | 2 | 3 | 3 | %100 |
| Database Creation | 2 | Database Design | Database Integration | 6 | 10 | 9 | %100 |
| Database  Integration | 3 | Database Creation  Back end registration | Testing | 19 | 21 | 0 | %0 |
| Testing | 4 | All Back end tasks | Deploy | 21 | 25 | 0 | %50 |

### Functional Decomposition

A functional breakdown to be used for scheduling is presented here.



### Task Network Diagram



### Timeline Chart

#### Project Schedule