payday

Team Rocket
Payday, Release 00.2 Alpha Web

System Test Plan
Revision 02

Specification Version: 1.0

Published: 2014

Author: Team Rocket

© Copyright Team Rocket 2014

Table of Contents

Payday, Release 00.2 Alpha Web

© Copyright Team Rocket 2014

© Copyright Team Rocket 2014

Payday, Release 00.2 Alpha Web

1.0 Business Objective

1.1 The Problem

Handling large amounts of business accounting and bookkeeping data is a cumbersome process for many small businesses and e-commerce retailers. Issues for multiple businesses include handling multiple employee management, employee payroll, revenue and expense transactions, and tracking sales over multiple periods of time. Many small businesses use low-tech spreadsheet bookkeeping management techniques resulting in poor records, loss of money, and loss of time.

1.2 The Solution

An ideal solution would allow an user-friendly web based software to automate bookkeeping, manage employees and track revenue and expense transactions. Payday is a web-based accounting and employee management system developed for small businesses. Payday shall perform common accounting functions and keep track of employees.

2.0 Client Requirements

2.1 Out of Scope Assumptions

1. Payday shall not provide support for multiple businesses or multiple managing users.
2. Payday shall not provide support for tax calculations based on business profits.
3. Payday shall not provide support for generating payment stubs, employee checks, and/or direct deposit on demand.

2.2 Requirements Traceability Matrix

1. Payday shall provide a login page for users, which shall query the database to check for correct login information.
   a) Use Case Name: UC_1_Login
   b) Type: SW
   c) Functional vs. Nonfunctional: Functional

2. Payday shall keep track of employee information.
   a) Use Case Name: UC_2_Employee_Information
   b) Type: SW
   c) Functional vs. Nonfunctional: Functional

3. Upon user request Payday shall generate expense reports based upon transactions stored in the database.
   a) Use Case Name: UC_3_Expense_Reports

© Copyright Team Rocket 2014

b) Type: SW
c) Functional vs. Nonfunctional: Functional

4. Payday shall provide functions for adding and deleting employee records. Payday will also provide functions to delete all records of a former employee. Payday will also notify the managing user of every creation, record date/time of employee creation, and list the employee under the main employee management dashboard pane.
   a) Use Case Name: UC_4_Edit_Employee_Functions
   b) Type: SW
   c) Functional vs. Nonfunctional: Functional

5. Payday shall provide the managing user an employee management dashboard pane with Employee listed by name.
   a) Use Case Name: UC_5_Manager_Pane
   b) Type: SW, SWC, DR
   c) Functional vs. Nonfunctional: Nonfunctional

6. Payday shall require at minimum the following employee information: Name, Address, Date of Birth, Social Security Number, Current Employment Status, and Job Title. Payday will prevent the creation of the employee record if the information asked has not been filled and provide the user with an "insufficient information" notification.
   a) Use Case Name: UC_6_New_Employee_Information
   b) Type: DWC, SW DR
   c) Functional vs. Nonfunctional: Nonfunctional

7. Payday shall provide functions for creating employee checks/pay stubs, tracking employee hours, editing employee history, and employee record viewing.
   a) Use Case Name: UC_7_Manage_Employee_Data
   b) Type: SW
   c) Functional vs. Nonfunctional: Functional

8. Payday shall provide the process of gathering the total amount of hours each employee has worked in a shift by getting the difference between his or her clock in and clock out time.
   a) Use Case Name: UC_8_Employee_Hours
   b) Type: SW
   c) Functional vs. Nonfunctional: Functional

9. Payday shall allow managers to be given access to alter/edit employee previous clock in and clock out times, number of hours worked by each employee, and quarterly employee pay stubs. Payday will also allow employees and managers to view a breakdown of each employee’s hourly work and payment history.
   a) Use Case Name: UC_9_Manager_Pay_Control
   b) Type: SW
   c) Functional vs. Nonfunctional: Functional

10. Payday shall allow the work history to be viewed as a past week, past 2 weeks, past month, or entire work history format.
    a) Use Case Name: UC_10_Pay_History
    b) Type: SWC
    c) Functional vs. Nonfunctional: Functional

11. Payday shall also calculate and allow managers to add revenue transaction and view daily, weekly, monthly, and yearly revenues.
    a) Use Case Name: UC_11_Revenue
    b) Type: SW
    c) Functional vs. Nonfunctional: Functional

12. Payday shall also take the revenue and expenses and allow the manager to view the information in a chart or graph format.
    a) Use Case Name: UC_12_RetrieveGraphData
    b) Type: SW, DR, NTW
    c) Functional vs. Nonfunctional: Functional

13. Payday shall allow the Managers and employees to change edit their personal user settings, including password, user ID, or profile image.
    a) Use Case Name: UC_13_EditUserSettings
    b) Type: SW
    c) Functional vs. Nonfunctional: Functional

14. Payday shall generate the net profit from the Profit & Loss statement of the business.
    a) Use Case Name: UC_14_ProfitLossStatement
    b) Type: SW, SWC
    c) Functional vs. Nonfunctional: Functional

3.0 Quality Assurance Guide for Test Case Review

3.1 Test Cases Review & Approval (to be done before testing):

Prepared By/Date:	**Reviewed By/Date:**	**Approved By/Date:**
Toufiq/16NOV2014	**Patel/16NOV2014**	**Byrd/16NOV2014**

3.2 Test Runs (to be filled during testing):

Run Number :	**1**	**2**	**3**
Conducted By:	Toufiq	Toufiq	Toufiq
Date :	16NOV2014	16NOV2014	16NOV2014

3.3 Defect Data Consolidation (to be filled after completion of testing):

Severity	**High**		**Medium**		**Low**		**Total**	
Type	**Syntax**	**Assignment**	**Interface**	**Checking**	**Function**	**Data**	**System**	**Documentation**
Cause	**TRANSACTION ERROR**				**USER-EDUCATION / TRAINING REQ'D**			

Phase of Origin							

3.4 Test Review and approval (to be filled after data consolidation):

Testing Reviewed By:	Date:	Testing Approved By:	Date:
Patel	16NOV2014	**Byrd**	16NOV2014

3.5 Test/QA Environment Information

Type	Server (Name, IP)	Schema or Link
Test Application Server	sheehantoufiq.com	http://sheehantoufiq.com/payday
Test Database Server	MYSQL	User: toufiq_payday Password: payday4350

3.6 Test/QA Credentials

Role	UserID	Password
User: Manager (Web Application)	manager@payday.com	test
User: Employee (Web Application)	*Manager must create employee as prerequisite. Manager shall determine UserID of employee.*	*Manager must create employee as prerequisite. Manager shall determine password of employee.*
User (MYSQL)	toufiq_payday	payday4350

3.7 Test Prerequisites:

This sub-section states all the key events that occurred before execution of testing.

- Brower homepage must be pointed to http://sheehantoufiq.com/payday.
- MYSQL Database must have executed schema payday_schema.sql.

3.8 Execution Details:

Execution details of testing depend on the application and can hence vary widely. The application could be a batch job, or an interactive one calling for on-line inputs, or a client-server job calling for a totally different approach. Given the wide diversity of tests, it is not advisable to impose a standard. However, it is suggested that regardless of nature of the testing, the following minimum information be recorded about the execution details of the testing:

- Behavior expected of the system
- Evaluation procedure, for interpreting the results
- Control decisions: When to start, end, suspend, resume, rerun the test; actions to be taken in case of exceptional situations
- Any other relevant aspect

If the test designer expects the test team to record or note down the behavior observed in a particular format, then he/she should provide samples of such forms as Annexures to the test cases, along with clear instructions on how the recording should be done.

3.9 Test Cases Statuses:

Pass: Expected and observed behavior is the same.
Fail: Expected and observed behavior is different. This is classified as a defect.

Defect Classifications:

- **H** - High severity defect
  Definition: System crash / Incorrect data which impacts other regions / etc.
- **M** - Medium severity defect
  Definition: Execution can continue / Incorrect data which does not impact other regions / etc.
- **L** - Low severity defect
  Definition: Cosmetic problem: Aesthetics / Message wording / Menu options / Wrong alarms / Help problems / etc.

4.0 Test Cases

4.1 Test Case - Login

Step No.	**Data Input**	**Expected Behavior**	**Observed Behavior ***					
			Run 1		**Run 2**		**Run 3**	
			(Pass / Fail)	**H/M/L**	**(Pass / Fail)**	**H/M/L**	**(Pass / Fail)**	**H/M/L**
1.	Navigate to Login Page	Payday's Login Page Displays	Pass	None	Pass	None	Pass	None
2.	Enter Manager email in email field	Manager email displays in email field	Pass	None	Pass	None	Pass	None
3.	Enter password in password field	Dots display in password field	Pass	None	Pass	None	Pass	None

© Copyright Team Rocket 2014

4.	Click Sign in	User is forwarded to Manager Dashboard	Pass	None	Pass	None	Pass	None

			Observed Behavior *					
Step No.	**Data Input**	**Expected Behavior**	**Run 1**		**Run 2**		**Run 3**	
			(Pass / Fail)	**H/M/L**	**(Pass / Fail)**	**H/M/L**	**(Pass / Fail)**	**H/M/L**
Prerequisites: At least one employee must exist in the database, created by a managing user.								
1.	Navigate to Login Page	Payday's Login Page Displays	Pass	None	Pass	None	Pass	None
2.	Enter Employee email in email field	Employee email displays in email field	Pass	None	Pass	None	Pass	None
3.	Enter password in password field	Dots display in password field	Pass	None	Pass	None	Pass	None
4.	Click Sign In	User is forwarded to Employee Dashboard	Pass	None	Pass	None	Pass	None

			Observed Behavior *					
Step No.	**Data Input**	**Expected Behavior**	**Run 1**		**Run 2**		**Run 3**	
			(Pass / Fail)	**H/M/L**	**(Pass / Fail)**	**H/M/L**	**(Pass / Fail)**	**H/M/L**
1.	Navigate to Login Page	Payday's Login Page Displays	Pass	None	Pass	None	Pass	None
2.	Enter User email in email field (incorrect)	User email displays in email field	Pass	None	Pass	None	Pass	None
3.	Enter password in password field	Dots display in password field	Pass	None	Pass	None	Pass	None
4.	Click Sign In	Red message displays stating that the user does not exist	Pass	None	Pass	None	Pass	None

			Observed Behavior *					
Step No.	**Data Input**	**Expected Behavior**	**Run 1**		**Run 2**		**Run 3**	
			(Pass / Fail)	**H/M/L**	**(Pass / Fail)**	**H/M/L**	**(Pass / Fail)**	**H/M/L**
1.	Navigate to Login Page	Login page displays	Pass	None	Pass	None	Pass	None
2.	Enter User email in email	User email displays in email	Pass	None	Pass	None	Pass	None

	field	field						
3.	Enter password in password field (incorrect)	Dots display in password field	Pass	None	Pass	None	Pass	None
4.	Click Sign In	Red message shows up stating that the password is incorrect	Pass	None	Pass	None	Pass	None

4.2 Test Case - Manage Employees: Create New Employee

Step No.	Data Input	Expected Behavior	Observed Behavior *					
			Run 1		Run 2		Run 3	
			(Pass / Fail)	H/M/L	(Pass / Fail)	H/M/L	(Pass / Fail)	H/M/L
Prerequisites: A managing user must be logged in.								
1.	Navigate to Employees Page	Employees Page displays	Pass	None	Pass	None	Pass	None
2.	Click "+" button	A "Create A Employee" popup modal displays	Pass	None	Pass	None	Pass	None
3.	Enter Employee's first name in first name field	Employee's first name displays in first name field	Pass	None	Pass	None	Pass	None
4.	Enter Employee's last name in last name field	Employee's last name displays in last name field	Pass	None	Pass	None	Pass	None
5.	Enter Employee’s phone number in phone number field	Employee's phone number displays in phone number field	Pass	None	Pass	None	Pass	None
6.	Enter Employee’s bio in bio field	Employee's bio displays in bio field	Pass	None	Pass	None	Pass	None
7.	Enter Employee’s wage in wage field	Employee's wage displays in wage field	Pass	None	Pass	None	Pass	None
8.	Enter Employee's email in email	Employee's email displays in email field	Pass	None	Pass	None	Pass	None

	field							
9.	Enter Employee’s password in password field	Dots display in password field	Pass	None	Pass	None	Pass	None
10.	Enter Employee’s password again in confirm password field	Dots display in confirm password field	Pass	None	Pass	None	Pass	None
11.	Click Create Employee	The popup modal disappears showing the new Employee added on the Employee's Page	Pass	None	Pass	None	Pass	None

Step No.	Data Input	Expected Behavior	Observed Behavior *					
			Run 1		Run 2		Run 3	
			(Pass / Fail)	H/M/L	(Pass / Fail)	H/M/L	(Pass / Fail)	H/M/L
Prerequisites: A managing user must be logged in.								
1.	Navigate to Employees Page	Employees Page displays	Pass	None	Pass	None	Pass	None
2.	Click "+" button	A "Create A Employee" popup modal displays	Pass	None	Pass	None	Pass	None
3.	Enter Employee's first name in first name field	Employee's first name displays in first name field	Pass	None	Pass	None	Pass	None
4.	Enter Employee's last name in last name field	Employee's last name displays in last name field	Pass	None	Pass	None	Pass	None
5.	Enter Employee’s phone number in phone number field	Employee's phone number displays in phone number field	Pass	None	Pass	None	Pass	None
6.	Enter Employee’s bio in bio field	Employee's bio displays in bio field	Pass	None	Pass	None	Pass	None
7.	Enter Employee’s wage in wage field	Employee's wage displays in wage field	Pass	None	Pass	None	Pass	None
8.	Enter	Employee's email	Pass	None	Pass	None	Pass	None

© Copyright Team Rocket 2014

	Employee's email in email field	displays in email field						
9.	Enter Employee’s password in password field	Dots display in password field	Pass	None	Pass	None	Pass	None
10.	Enter Employee’s password again in confirm password field (incorrect)	Dots display in confirm password field. Red message displays stating that the passwords have to be the same	Pass	None	Pass	None	Pass	None

4.3 Test Case - Manage Employees: View All Employees

Step No.	**Data Input**	**Expected Behavior**	**Observed Behavior ***					
			Run 1		**Run 2**		**Run 3**	
			(Pass / Fail)	**H/M/L**	**(Pass / Fail)**	**H/M/L**	**(Pass / Fail)**	**H/M/L**
Prerequisites: At least one employee must exist in the database, created by a managing user.								
1.	Navigate to Employees Page	All existing employees are displayed	Pass	None	Pass	None	Pass	None

4.4 Test Case - Manage Employees: View Employee Information

Step No.	**Data Input**	**Expected Behavior**	**Observed Behavior ***					
			Run 1		**Run 2**		**Run 3**	
			(Pass / Fail)	**H/M/L**	**(Pass / Fail)**	**H/M/L**	**(Pass / Fail)**	**H/M/L**
Prerequisites: At least one employee must exist in the database, created by a managing user.								
1.	Navigate to Employees Page	Employees Page displays	Pass	None	Pass	None	Pass	None
2.	Click on existing	Employee’s Information	Pass	None	Pass	None	Pass	None

© Copyright Team Rocket 2014

	Employee	Page displays						

4.5 Test Case - Manage Employees: Update Employee

Step No.	**Data Input**	**Expected Behavior**	**Observed Behavior ***					
			Run 1		**Run 2**		**Run 3**	
			(Pass / Fail)	**H/M/L**	**(Pass / Fail)**	**H/M/L**	**(Pass / Fail)**	**H/M/L**
Prerequisites: At least one employee must exist in the database, created by a managing user.								
1.	Navigate to Employees Page	Employees Page displays	Pass	None	Pass	None	Pass	None
2.	Click on existing Employee	Employee’s Information Page displays	Pass	None	Pass	None	Pass	None
3.	Click “Edit” button	Editable regions show	Pass	None	Pass	None	Pass	None
4.	Perform necessary edits into inputs	User edits show in inputted fields	Pass	None	Pass	None	Pass	None
5.	Click “Save” button	Editable regions disappear, the User edits have been saved successfully	Pass	None	Pass	None	Pass	None
Step No.	**Data Input**	**Expected Behavior**	**Observed Behavior ***					
			Run 1		**Run 2**		**Run 3**	
			(Pass / Fail)	**H/M/L**	**(Pass / Fail)**	**H/M/L**	**(Pass / Fail)**	**H/M/L**
Prerequisites: At least one employee must exist in the database, created by a managing user.								
1.	Navigate to Employees Page	Employees Page displays	Pass	None	Pass	None	Pass	None
2.	Click on existing Employee	Employee’s Information Page displays	Pass	None	Pass	None	Pass	None
3.	Click “Edit” button	Editable regions show	Pass	None	Pass	None	Pass	None
4.	Perform necessary edits into inputs (leave a required field blank)	User edits show in inputted fields	Pass	None	Pass	None	Pass	None

5.	Click “Save” button	A notification displays stating that the required inputs are blank	Pass	None	Pass	None	Pass	None

4.6 Test Case - Manage Employees: Delete Employee

Step No.	Data Input	Expected Behavior	Observed Behavior *					
			Run 1		Run 2		Run 3	
			(Pass / Fail)	H/M/L	(Pass / Fail)	H/M/L	(Pass / Fail)	H/M/L
Prerequisites: At least one employee must exist in the database, created by a managing user.								
1.	Navigate to Employees Page	Employees Page displays	Pass	None	Pass	None	Pass	None
2.	Click on existing Employee	Employee’s Information Page displays	Pass	None	Pass	None	Pass	None
3.	Click “Delete” button	A prompt appears asking “Are you sure you would like to delete this employee?” with a “Cancel” and “Delete” button	Pass	None	Pass	None	Pass	None
4	Click “Delete” Button	The user is redirected to the Employees page, the employee has been deleted, and no longer appears	Pass	None	Pass	None	Pass	None
Step No.	**Data Input**	**Expected Behavior**	**Observed Behavior ***					
			Run 1		**Run 2**		**Run 3**	
			(Pass / Fail)	**H/M/L**	**(Pass / Fail)**	**H/M/L**	**(Pass / Fail)**	**H/M/L**
Prerequisites: At least one employee must exist in the database, created by a managing user.								
1.	Navigate to Employees Page	Employees Page displays	Pass	None	Pass	None	Pass	None
2.	Click on existing Employee	Employee’s Information Page displays	Pass	None	Pass	None	Pass	None
3.	Click “Delete”	A prompt appears asking “Are you	Pass	None	Pass	None	Pass	None

	button	sure you would like to delete this employee?” with a “Cancel” and “Delete” button						
4	Click “Cancel” Button	The user is redirected to the Employee’s Information Page. No changes have occurred.	Pass	None	Pass	None	Pass	None

4.7 Test Case - Edit User Settings

Step No.	Data Input	Expected Behavior	Observed Behavior *					
			Run 1		Run 2		Run 3	
			(Pass / Fail)	H/M/L	(Pass / Fail)	H/M/L	(Pass / Fail)	H/M/L
1.	Navigate to “User Settings” Page	User Setting Page displays	Pass	None	Pass	None	Pass	None
2.	Click “Edit” button	Editable regions show	Pass	None	Pass	None	Pass	None
3.	Perform necessary edits into inputs	User edits show in inputted fields	Pass	None	Pass	None	Pass	None
4.	Click “Save” button	Editable regions disappear, the User edits have been saved successfully	Pass	None	Pass	None	Pass	None
Step No.	**Data Input**	**Expected Behavior**	**Observed Behavior ***					
			Run 1		**Run 2**		**Run 3**	
			(Pass / Fail)	**H/M/L**	**(Pass / Fail)**	**H/M/L**	**(Pass / Fail)**	**H/M/L**
1.	Navigate to “User Settings” Page	User Setting Page displays	Pass	None	Pass	None	Pass	None
2.	Click “Edit” button	Editable regions show	Pass	None	Pass	None	Pass	None
3.	Perform necessary edits into	User edits show in inputted fields	Pass	None	Pass	None	Pass	None

© Copyright Team Rocket 2014

	inputs (leave a required field blank)							
4.	Click “Save” button	A notification displays stating that the required inputs are blank	Pass	None	Pass	None	Pass	None

4.8 Test Case - Manage Transactions: Create New Transaction

Step No.	Data Input	Expected Behavior	Observed Behavior *					
			Run 1		Run 2		Run 3	
			(Pass / Fail)	H/M/L	(Pass / Fail)	H/M/L	(Pass / Fail)	H/M/L
1.								
2.								
3.								
Step No.	Data Input	Expected Behavior	Observed Behavior *					
			Run 1		Run 2		Run 3	
			(Pass / Fail)	H/M/L	(Pass / Fail)	H/M/L	(Pass / Fail)	H/M/L
1.								
2.								
3.								

4.9 Test Case - Manage Transactions: View All Transactions

Step No.	Data Input	Expected Behavior	Observed Behavior *					
			Run 1		Run 2		Run 3	
			(Pass / Fail)	H/M/L	(Pass / Fail)	H/M/L	(Pass / Fail)	H/M/L
1.								
2.								
3.								
Step No.	Data Input	Expected Behavior	Observed Behavior *					
			Run 1		Run 2		Run 3	
			(Pass / Fail)	H/M/L	(Pass / Fail)	H/M/L	(Pass / Fail)	H/M/L

© Copyright Team Rocket 2014

1.								
2.								
3.								

4.10 Test Case - Manage Transactions: Update Transaction

Step No.	Data Input	Expected Behavior	Observed Behavior *					
			Run 1		Run 2		Run 3	
			(Pass / Fail)	H/M/L	(Pass / Fail)	H/M/L	(Pass / Fail)	H/M/L
1.								
2.								
3.								
Step No.	Data Input	Expected Behavior	Observed Behavior *					
			Run 1		Run 2		Run 3	
			(Pass / Fail)	H/M/L	(Pass / Fail)	H/M/L	(Pass / Fail)	H/M/L
1.								
2.								
3.								

4.11 Test Case - Manage Transactions: Delete Transaction

Step No.	Data Input	Expected Behavior	Observed Behavior *					
			Run 1		Run 2		Run 3	
			(Pass / Fail)	H/M/L	(Pass / Fail)	H/M/L	(Pass / Fail)	H/M/L
1.								
2.								
3.								
Step No.	Data Input	Expected Behavior	Observed Behavior *					
			Run 1		Run 2		Run 3	
			(Pass / Fail)	H/M/L	(Pass / Fail)	H/M/L	(Pass / Fail)	H/M/L
1.								
2.								
3.								

5.0 Sign-off

Signature	Name & Title	Date	Comments
	Sheehan Toufiq	16-11-2014	
	Justin Patel	16-11-2014	
	Benjamin Byrd	16-11-2014	
	Andrew Rutherford	16-11-2014	
	Suraj Sequeira	16-11-2014	

© Copyright Team Rocket 2014

6.0 Appendices

6.1 References

- Document Store:
  /directory/document-store/

- Application URL:
  http://sheehantoufiq.com/payday

- Code Repository:
  https://github.com/bebyrd/payday

6.2 Supplementary Attachments or Samples

Attached Items:

- Product Blurb:
  Location:
  /directory/document-store/

- Requirements Traceability Matrix
  Location
  /directory/document-store/

- Use Case Presentation and Screenshots
  Location:
  /directory/document-store/

- Individual Use Cases
  Location:
  /directory/document-store/

- User Interaction Diagram
  Location:
  /directory/document-store/

- Initial Mockups
  Location:
  /directory/document-store/

© Copyright Team Rocket 2014