

Team Name: NightOwl

Membership: Jarvis Kampe, Hieu Nguyen, David Piholyuk, Derek Ruiz-Garcia, Paul Schmidt

Website URL: <https://github.com/NightOwlTCSS360>

Contact Information: nightowlcss360@gmail.com

Table of Contents

Table of Contents.....	1
Introduction.....	3
Requirements Addressed.....	4
User Stories.....	4
USX01: YES.....	4
USX02: YES.....	4
USX03: YES.....	4
USX04: YES.....	4
USX05: YES.....	4
US01: (Partial Implementation).....	4
US02: YES.....	4
US03: NO.....	4
US04: YES.....	4
US05: NO.....	5
US06: NO.....	5
US07: NO.....	5
US08: NO.....	5
US09: NO.....	5
US10: NO.....	5
US11: NO.....	5
US12: NO.....	5
Business Rules.....	5
BR01: YES.....	5
Technical Requirements.....	6
TRX01: YES.....	6
Non-Functional Requirements.....	6
NF01: FULFILLED.....	6
NF02: FULFILLED.....	6
NF03: FULFILLED.....	6
NF04: FULFILLED.....	6
NF05: FULFILLED.....	6
NF06: FULFILLED.....	6
NF07: FULFILLED.....	6
NF08: FULFILLED.....	7

NF09: FULFILLED.....	7
NF10: FULFILLED.....	7
NF11: FULFILLED.....	7
Running the Application.....	8
Contributions.....	10
Build and Tests.....	15
Source Catalog.....	16
Emergency Contacts.....	19

Introduction

Project Budgeting System by Nightowl is a desktop application that helps the user plan out and track the progress of their personal projects. Supporting multiple-user logins, relevant information and project data is saved and stored on the user's device locally. Users also have the ability to export and import projects out of and into their list of projects.

Users can create multiple projects each with their own estimated budgets set by the user. Each project also has a cumulative projected cost which is calculated by summing the costs of all purchases the user has associated with the project. The difference between the estimated budget and costs helps the user identify whether the project is on track to stay under budget or if the projected costs of the project will surpass the budget allowed. Users can track the progress of their project by confirming purchases are made, which will update the status of the Tasks associated with the project and ultimately the status of the project itself.

Requirements Addressed

User Stories

USX01: YES

As a user I want to set up the app so it will have my first name and email address.

USX02: YES

As a user I want to see an About screen for the app so I can see the version number.

USX03: YES

As a user I would like to export settings for synchronization with other devices.

USX04: YES

As a user I would like to import settings to synchronize with other devices.

USX05: YES

As a user I would like to export and import other information in the application.

US01: (Partial Implementation)

As a user, I would like to be able to create a project with estimates for materials and time because it allows me to plan ahead for a project.

(Estimates for time not implemented, but estimates for costs are)

US02: YES

As a user, I would like to be able to journal the progress of the project, record expenses, and track them against the projected budget so I can see how the project is developing.

US03: NO

As a user, I would like to be able to organize project files and plans by keeping track of receipts, installation information, warranty information, and sketches.

US04: YES

As a user, I would like to be able to know when expenses exceed the budget, or when savings would allow for more expensive choices.

US05: NO

As a user, I would like to be able to store my projects in the cloud so that I can access them from anywhere.

US06: NO

As a user, I would like to be able to switch between a “view” and “edit” mode so that I can allow other users to view the project but not edit it.

US07: NO

As a user, I would like to be able to discard any documentation that I deem unnecessary.

US08: NO

As a user, I would like to be able to use a keyword or tag to quickly locate information like instructions, the warranty, the receipt, my drawings, and the website links.

US09: NO

As a user, I would like to be able to access the application from my desktop, phone, and/or tablet.

US10: NO

As a user, I would like to be notified when a warranty will be expiring soon.

US11: NO

As a user, I would like to be able to scan in my personal drawings and sketches for a project and reference them later.

US12: NO

As a user, I would like to be able to save the information from a website in case the website becomes unavailable in the future.

Business Rules

BR01: YES

The application must support account creation.

Technical Requirements

TRX01: YES

The version number cannot be hard coded in the UI – The version number must be requested by the UI code from a constant property of an object. The code for the object may change from version to version, the UI code must not.

Non-Functional Requirements

NF01: FULFILLED

Other than features that involve refreshing or uploading data, all aspects of the application must be able operate offline, meaning there is no connection to a network or the Internet.

NF02: FULFILLED

The system does not require secure user authentication. Where an identity is needed, a person signs in when the application starts using a unique name and or PIN.

NF03: FULFILLED

The application executes from a runnable jar file on a single Windows machine as a desktop application with only a single user at a time.

NF04: FULFILLED

Persistent data will be stored in one or more files on the local file system.

NF05: FULFILLED

The programming will be carried out using current versions of Eclipse or equivalent IDE, Java, and JUnit.

NF06: FULFILLED

Source code will be managed in a GitHub repository.

NF07: FULFILLED

All third-party components and libraries must be approved in advance and properly cited. Third party executable components must be packaged with the application and cannot depend on prior installation or configuration.

NF08: FULFILLED

There is a consistent look among the different screens.

NF09: FULFILLED

On each screen, users have the choice to go to the previous screen and the main menu (among any other options you might provide).

NF10: FULFILLED

Users are unaware of the location of the files that hold the persistent data. No part of the interface should cause users to deal with specifying the location. Instead, the location of these files should be initialized by your program. A common technique is to use an initialization or configuration file that is processed at startup.

NF11: FULFILLED

You may assume that users know how to navigate a file system with an explorer style or finder style browser.

Running the Application

NOTE: The JAR is compiled with JDK 19. Make sure to have an up to date version of the Java Runtime Environment (19+) installed in order to mitigate issues running the program.

NOTE: The program will create a directory to store local persistent data by using your User Home directory. If you wish to remove the data after testing, make sure to remove the directory from your computer.

For example: C:\Users\Name\ProjectBudgetingSystem.

1. Double-Click the JAR file to start up the program.
2. Click the “Sign Up” button and register a new user. Input validation is required during the signup process, so make sure you use proper conventions. Here is the input you can use if you wish: [Email = bob@uw.edu, FirstName = Bob, LastName = McBob, Password = NightOwl]
3. You can now login with the email and password you used to sign up. Enter in the relevant email and password, and click the “Sign In” button.
4. You can create a new project by clicking the plus-in-a-circle icon next to the Projects List panel. Enter in a name for a new project such as “Deck” and hit okay to confirm.
5. You will be prompted to enter a budget for the project. Enter “1000” and click the confirmation button.
6. The Project should be automatically selected in the Projects List. You can now add a Task to the Project by clicking the plus-in-a-circle icon next to the Tasks List header. Enter in a Task name such as “Buy Lumber”.
7. The new task should be automatically selected. Now you can add a new Purchase to the Task. To do so, click the plus-in-a-circle icon next to the title “Purchases”. Add a name for this Purchase such as “Planks” and a cost such as “350.50”. Confirm the creation.
8. After making a change to the budget or adding/deleting/editing purchases/task, make sure to click the “Recalculate” button to see the updated information in the budget panel.
9. If you wish to add more purchases to this task, you may by completing step 6 again. If you wish to add more Tasks and purchases associated with those tasks you can do so by completing steps 7-8 again. If you wish to add additional projects with their own tasks and purchases, do so by completing steps 4-8 again.

10. You can navigate between different projects and tasks by clicking on the Panels generated for those respective projects/tasks. You can visually see which Project/Task is selected because the panel you click will become dark gray.
11. To track the progress of a project, click the check marks associated with each purchase to signify that they are complete. Once all purchases have been completed for a task, the check mark next to the task name will become checked.
12. To view the About Panel, while logged in select the Help dropdown menu and click the About MenuItem. To go back to the Project view, click the back button.
13. To sign out, select the Sign Out option from the Settings dropdown menu.
14. To export a project, select the Export Projects button from the File dropdown menu. Select the Project you wish to export from the populated list of Projects. Choose the destination folder for the exported project via the “...” button. After selecting a project and destination, the “Export” button will become enabled. Click this button to Export the project.
15. To import a project, select the Import Projects button from the File dropdown menu. Navigate to the .ser file you wish to import, select it, and confirm. If the import is successful you will see a new Project populated in the Projects List.

Contributions

Developer	Class/Method	Activity(Coding/Editing)
David Piholyuk	View	Coding
David Piholyuk	LoginPanel	Coding
David Piholyuk	LogoPanel	Coding
David Piholyuk	ProjectPanel	Coding
David Piholyuk	SignUpPanel	Coding
David Piholyuk	TaskPanel	Coding
David Pihoyuk	pom.xml	Editing
Derek Jagger Ruiz Garcia	PDC/createSettingsFile	Coding
Derek Jagger Ruiz Garcia	PDC/importSettings	Coding
Derek Jagger Ruiz Garcia	PDC/exportSettings	Coding
Derek Jagger Ruiz Garcia	PDC/addNewPurchase	Coding
Derek Jagger Ruiz Garcia	PDC/editPurchaseCost	Coding
Derek Jagger Ruiz Garcia	PDC/deletePurchase	Coding
Derek Jagger Ruiz Garcia	PDC/addNewTask	Coding
Derek Jagger Ruiz Garcia	PDC/deleteCurrentTask	Coding
Derek Jagger Ruiz Garcia	PDC/isNonNegativeDouble	Coding
Derek Jagger Ruiz Garcia	Task/deletePurchase	Coding
Derek Jagger Ruiz Garcia	ProjectPanel/Constructor	Editing
Derek Jagger Ruiz Garcia	ProjectPanel/repopulatePurchasesList	Editing
Derek Jagger Ruiz Garcia	ProjectPanel/repopulateTasksList	Coding
Derek Jagger Ruiz Garcia	ProjectPanel/propertyChan	Coding

	ge	
Derek Jagger Ruiz Garcia	ProjectPanel/addPropertyChangeListener	Coding
Derek Jagger Ruiz Garcia	ProjectPanel/TaskAddActionPerformed	Coding
Derek Jagger Ruiz Garcia	ProjectPanel/PurchaseAddActionPerformed	Coding
Derek Jagger Ruiz Garcia	ProjectPanel/TrashButtonActionPerformed	Coding
Derek Jagger Ruiz Garcia	TaskPanel/PropertyChange	Coding
Derek Jagger Ruiz Garcia	PurchasePanel	Coding
Derek Jagger Ruiz Garcia	PurchasePopUpPanel	Coding
Derek Jagger Ruiz Garcia	ProjectTest	Coding
Derek Jagger Ruiz Garcia	PurchaseTest	Coding
Derek Jagger Ruiz Garcia	TaskTest	Coding
Paul Schmidt	PDC/Constructor	Coding
Paul Schmidt	PDC/logoutUser	Coding
Paul Schmidt	PDC/addNewProject	Coding
Paul Schmidt	PDC/exportProject	Coding
Paul Schmidt	PDC/importProject	Coding
Paul Schmidt	PDC/deleteCurrentProject	Coding
Paul Schmidt	ProjectManagerMain	Coding
Paul Schmidt	Project	Coding
Paul Schmidt	Task	Coding
Paul Schmidt	Purchase	Coding
Paul Schmidt	AppInfo	Coding
Paul Schmidt	Developer	Coding

Paul Schmidt	User/Constructor	Coding/Editing
Paul Schmidt	User/loadInUserProjects	Coding
Paul Schmidt	User/addProject	Coding
Paul Schmidt	ExportDialog	Coding (w/ NetBeans IDE)
Paul Schmidt	AboutPanel/Constructor	Coding
Paul Schmidt	AboutPanel/jButton1Action Performed	Editing
Paul Schmidt	DashboardPanel/repopulat eProjectPanel	Coding
Paul Schmidt	DashboardPanel/repopulat eProjectList	Coding
Paul Schmidt	ProjectList/repopulateProje ctsList	Coding
Paul Schmidt	ProjectPanel/ProjectTrash Button	Coding
Paul Schmidt	ProjectsPanel/setCurrentPr ojectAction	Coding
Paul Schmidt	View/importProjectAction	Coding
Paul Schmidt	View/exportProjectAction	Coding
Paul Schmidt	View/setMenuBarVisibility	Coding
Paul Schmidt	View/setDashboardPanel	Coding
Paul Schmidt	PDCImportExportProjectsT est	Coding
Paul Schmidt	PDCMainTest	Coding
Paul Schmidt	PDCSettingsTest	Coding
Paul Schmidt	pom.xml /build	Coding/Editing
Hieu Nguyen	PDC/totalCost	Coding
Hieu Nguyen	PDC/addProjectCurrentUs er	Coding

Hieu Nguyen	PDC/calculateCurrentBudget	Coding
Hieu Nguyen	CvsHandler/readAll	Coding
Hieu Nguyen	CSVHandler/LoadUser	Coding
Hieu Nguyen	CSVHandler/writeAll	Coding
Hieu Nguyen	UserController/ addUser	Coding
Hieu Nguyen	UserController/ Login	Coding
Hieu Nguyen	UserController/ UserController	Coding
Hieu Nguyen	AddBudgetPopUpPanel	Coding
Hieu Nguyen	BudgetPanel	Coding
Hieu Nguyen	DashBoard/ createProjectAction	Coding/Editing
Hieu Nguyen	DashBoardPanel/repopulateProjectPanel()	Editing
Hieu Nguyen	LoginPanel/jButton2.addActionListener	Coding/Editing
Hieu Nguyen	SignUpPanel/jButton1.addActionListener	Coding/Editing
Hieu Nguyen	ProjectPanel/repopulateBudgetInfo	Coding
Hieu Nguyen	ProjectList/createProjectButtoncreateProjectAction	Coding/Editing
Jarvis Kampe	AppInfo	Coding/Editing
Jarvis Kampe	AboutPanel	Coding
Jarvis Kampe	DashboardPanel	Coding
Jarvis Kampe	ProjectList	Coding

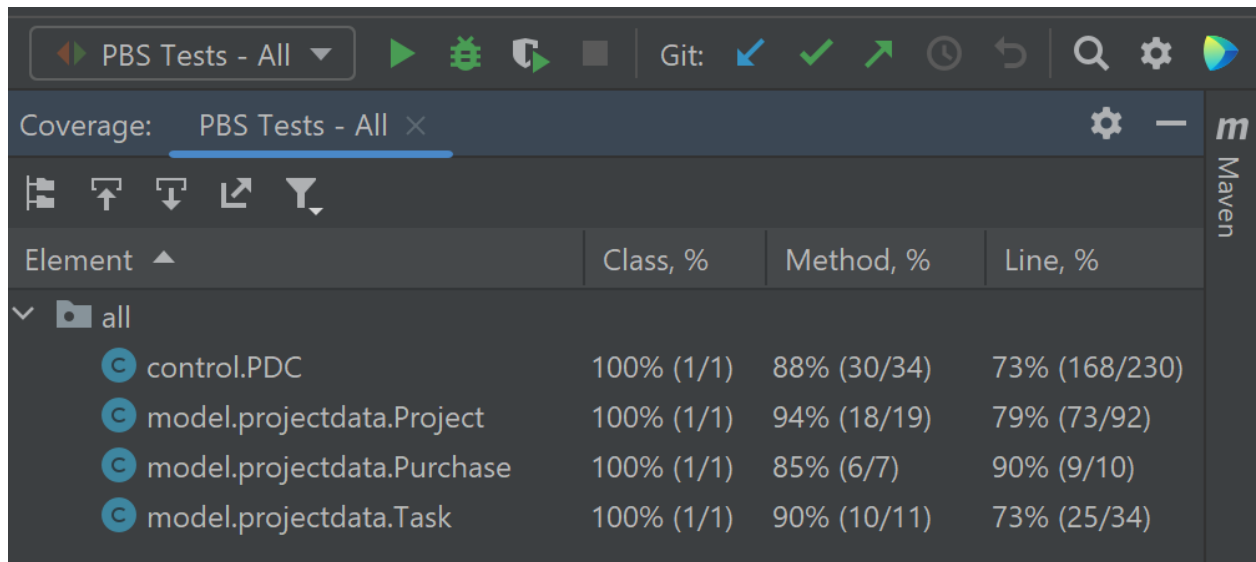
Jarvis Kampe	ProjectsPanel	Coding
Jarvis Kampe	View	Coding/Editing

Build and Tests

Link to an example BuildLog generated during production: [Build Log](#)

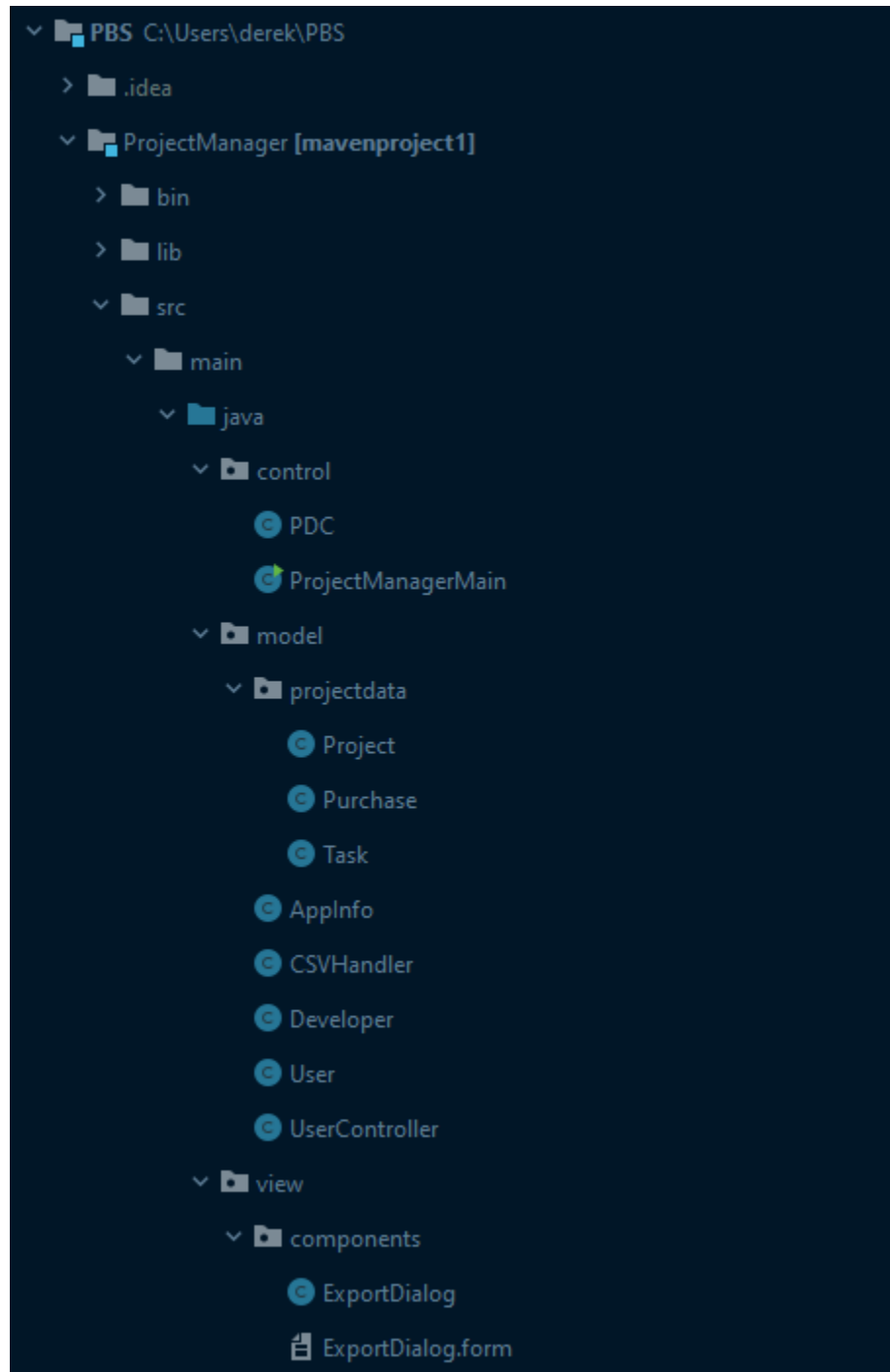
The JAR is built using JDK 19, so you must make sure that the JRE you use to run the program is a current enough version.


























Test Coverage








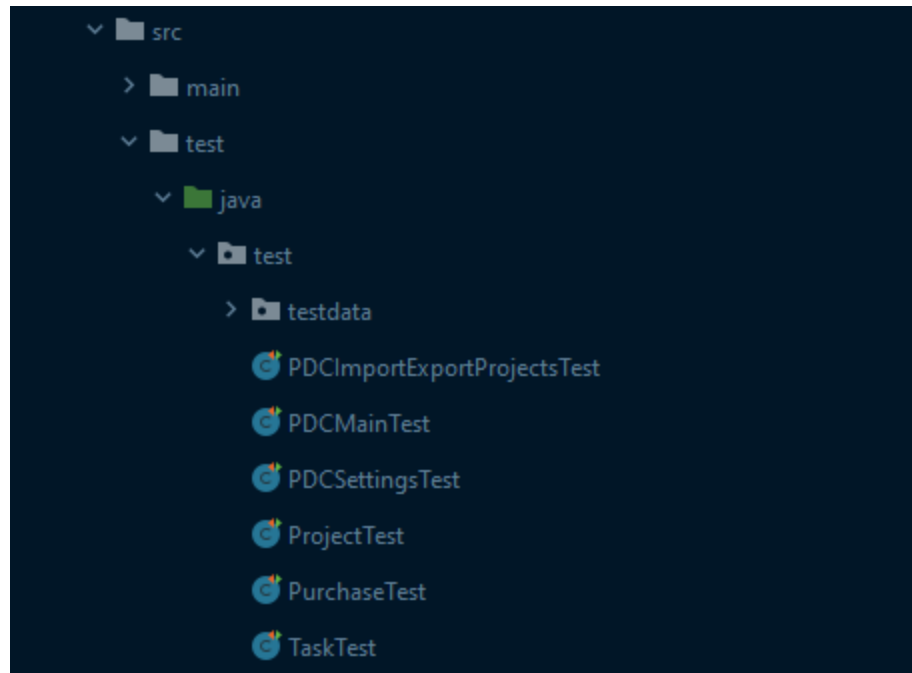
Element ▲	Class, %	Method, %	Line, %
▼ all			
control.PDC	100% (1/1)	88% (30/34)	73% (168/230)
model.projectdata.Project	100% (1/1)	94% (18/19)	79% (73/92)
model.projectdata.Purchase	100% (1/1)	85% (6/7)	90% (9/10)
model.projectdata.Task	100% (1/1)	90% (10/11)	73% (25/34)

Source Catalog



-  About
-  About.form
-  AboutPanel
-  AboutPanel.form
-  AddBudgetPopUpPanel
-  BudgetPanel
-  Dashboard.form
-  DashboardPanel
-  DashboardPanel.form
-  GuestRegistration.form
-  LoginPanel
-  LoginPanel.form
-  LogoPanel
-  LogoPanel.form
-  ProjectList
-  ProjectList.form
-  ProjectPanel
-  ProjectPanel.form
-  ProjectsPanel
-  ProjectsPanel.form
-  PurchasePanel
-  PurchasePanel.form
-  PurchasePopUpPanel
-  PurchasePopUpPanel.form
-  SignUpPanel

-  SignUpPanel.form
-  TaskPanel
-  TaskPanel.form
-  View
-  View.form



Emergency Contacts

Contact 1:

Paul Schmidt - schmipd@uw.edu

Contact 2:

Derek J. Ruiz Garcia - djruiz49@uw.edu