

State Park Payment Project Memo

Project status: completed


























Repository: https://bitbucket.org/Wayne_Song/cs445-ood/src/master/Project/

Lines of code (Measured by Metrics -Eclipse's Plugin)

▼ Total Lines of Code	3061
▼ src	3061
> edu.iit.cs445.StateParking.Objects	1320
> edu.iit.cs445.StateParking.REST	976
> edu.iit.cs445.StateParking.UnitTest	707
> edu.iit.cs445.StateParking.ObjectsEnum	33
> edu.iit.cs445.StateParking.Actors	25

Unit test Coverage:

Tool: Junit4. Coverage measured by EclEmma – Eclipse's Plugin. The coverage rate for Objects and ObjectsEnum packages are 92.9% and 100% respectively.

▼ edu.iit.cs445.StateParking.Objects		92.9 %	2,506	192
> MockDatabase.java		89.0 %	824	102
> ListOfPark.java		72.5 %	129	49
> Park.java		88.6 %	265	34
> ListOfNote.java		96.9 %	157	5
> ListOfOrder.java		98.7 %	155	2
> Address.java		100.0 %	67	0
> CreditCard.java		100.0 %	34	0
> geo.java		100.0 %	49	0
> ListOfVisitor.java		100.0 %	107	0
> MockCCprocessor.java		100.0 %	89	0
> NoteObj.java		100.0 %	108	0
> NullNote.java		100.0 %	25	0
> NullOrder.java		100.0 %	26	0
> NullPark.java		100.0 %	39	0
> NullVisitor.java		100.0 %	21	0
> OrderObj.java		100.0 %	128	0
> Report.java		100.0 %	43	0
> ReportCode.java		100.0 %	15	0
> ReportOfAdmission.java		100.0 %	55	0
> RevenueReport.java		100.0 %	69	0
> Vehicle.java		100.0 %	41	0
> VisitorObj.java		100.0 %	60	0
> edu.iit.cs445.StateParking.UnitTest		96.1 %	2,286	93
> edu.iit.cs445.StateParking.ObjectsEnum		100.0 %	712	0

Cyclomatic Complexity:

The Cyclomatic complexity is measured by Metrics Plug-in in Eclipse IDE.

Average Per Method: 2.194

Maximum Cyclomatic complexity: 12 (a method for finding note/order within a given time range)

Number of Hours needed for coding: about 80 hours

Number of Hours needed for preparing submission: about 6 hours

List of challenges:

- How to delete object without affecting objects associated with it?
A: Using null object pattern and map the IDs of associated objects together. When trying to access an ID of a deleted object, a null object is returned for exception handling.
- How to make a deployable war file?
A: To deploy a war file to Tomcat the war file needs to consist of 2 directories META-INF and WEB-INF. The classes files of project will sit in the "classes" folder inside WEB-INF.

Recommendation

- Release the API and required output earlier, some instructions about parsing nested class to given JSON format will also be helpful.
- Tutorial about how to generate and deploy war file to Tomcat from java project using Ubuntu terminal. The sample project tutorial was deployed using ubuntu GUI. If we can add a second part about how to generate, deploy war file to Tomcat server and make http request without using GUI (only in Ubuntu terminal), it will be more helpful in my opinion.

Extra Implementation

- A mocked credit card processor that detect 3 card types (VISA, Master Card and American Express as described in the API), check if the expiration date is in the right format and if the card has expired.
- Delete note function.

To do:

- 1) Create different actors. Each actor only has access to part of the methods, instead of all of them. Move half of the methods in MockDatabase class to different actors.
- 2) Log-in system for different actors

License

- ParkAdmin is an open-sourced software licensed under CC By 4.0

Known Bug

- TestSuit is sensitive. Changing order might cause failure. (Individual test is not affected)
- Address format is required to be “(Street), (City), (State)(\space)(Zip Code)”. Using a different format for data post might end up http 400 response indicating wrong address format.
- Some fields of admission /revenue reports’ out put json might have slightly different names (like StartDate instead of start_date) comparing to required output.

Credits and acknowledgements

- 1. Jersey - <https://jersey.java.net/>
- 2. Gson - <https://github.com/google/gson>
- Virgil Bistriceanu, M.S., Adjunct Faculty, Computer Science, Illinois Institute of Technology.
- Xintian Lee, M.S., Computer Science, Illinois Institute of Technology