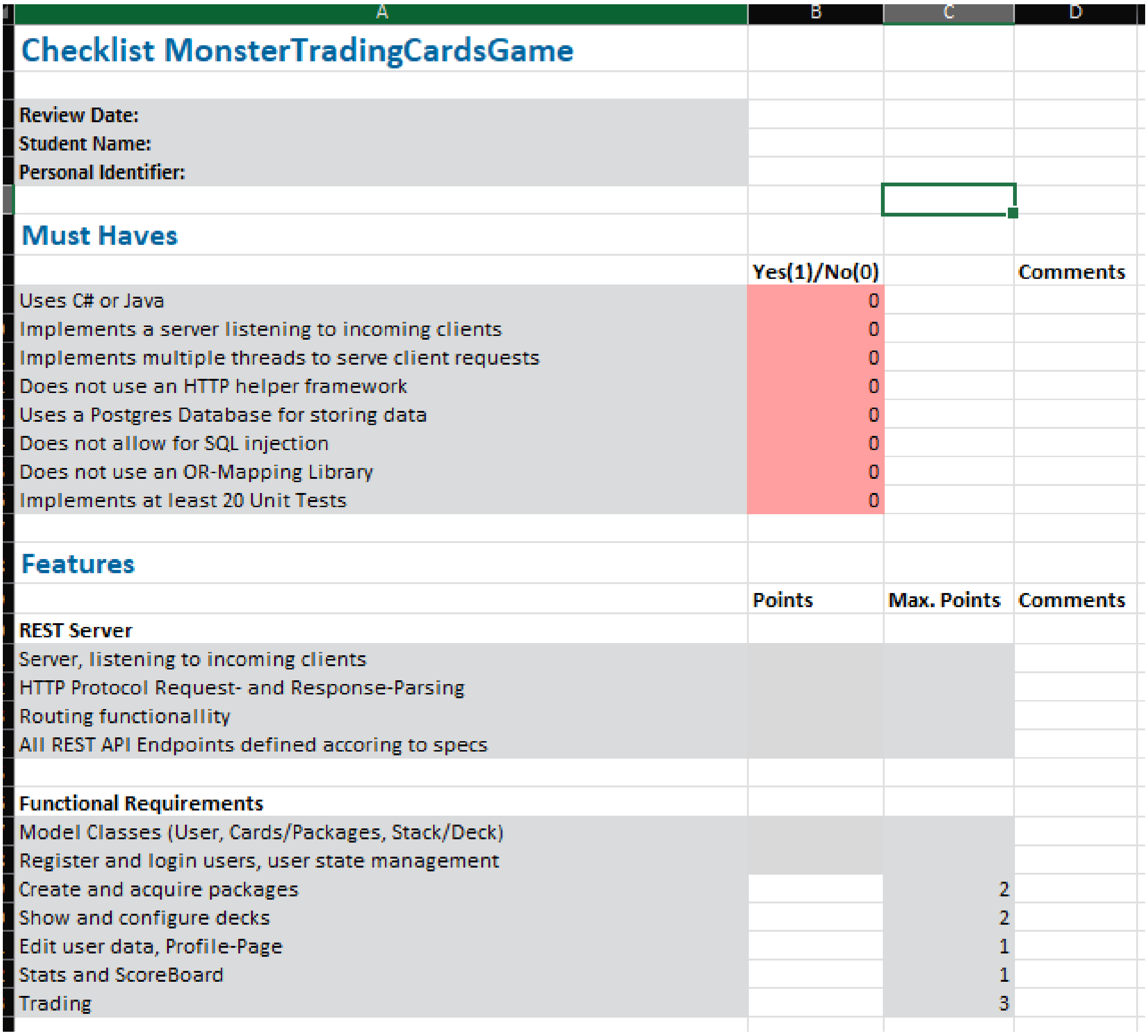

Checklist MonsterTradingCardsGame

Review Date:
Student Name:
Personal Identifier:

Must Haves

	Yes(1)/No(0)		Comments
Uses C# or Java	0		
Implements a server listening to incoming clients	0		
Implements multiple threads to serve client requests	0		
Does not use an HTTP helper framework	0		
Uses a Postgres Database for storing data	0		
Does not allow for SQL injection	0		
Does not use an OR-Mapping Library	0		
Implements at least 20 Unit Tests	0		

Features

	Points	Max. Points	Comments
REST Server			
Server, listening to incoming clients			
HTTP Protocol Request- and Response-Parsing			
Routing functionallity			
All REST API Endpoints defined accoring to specs			
Functional Requirements			
Model Classes (User, Cards/Packages, Stack/Deck)			
Register and login users, user state management			
Create and acquire packages		2	
Show and configure decks		2	
Edit user data, Profile-Page		1	
Stats and ScoreBoard		1	
Trading		3	

Battle		
Battle-Logic, Play rounds		4
Draw possible		2
Clean log of battle		3
take-over cards after loss of a round		2
consider specialities in battle-rounds between cards		1
ELO calculation		2
Mandatory Unique Feature		3
Non-functional Requirements		
Token-based security		
Persistence (DB)		8
Quality of unit-tests (usefulness, no duplicates, ...)		4
Integration Tests (curl or alternatively a custom app, working automatically)		2
Bonus Features		
Protocol		
Describes app design (decisions, structure, class diagrams)		0,5
Describes lessons learned		1
Describes unit testing decisions		1
Describes unique feature		1
Contains tracked time		0,5
Contains link to GIT		
Sum Points	**0**	**44**