

Unit Tests

Unit tests for database

Precondition: An in-memory database is created

Postcondition: The in-memory database created in the Pre-condition should be closed and deleted

ID	Test Case Description	Precondition	Input	Expected Output	Actual Output	Test Result	Test Comments
1	AddCardEntity: Insert a new card entity to the database. The card should be inserted into the database when the correct Dao method is called	There is a CardEntity object	cardEntityDao.insert(cardEntity)	The card entity is inserted	The card entity is inserted	PASS	
2	SearchCardByName: Search cards that has the given name. It should return a list of cards with the given name, or just an empty list if there is no such cards. The search should be case insensitive	There is a CardEntity object with the name "sam" in the database	val cards = cardEntityDao.getCardsByName("Sam");	cards contains and only contains card entity objects with name "sam" (case insensitive)	cards contains and only contains one card entity objects with name "sam"	PASS	
3	GetAllCardsOrderByName: Order cards entity object by name. It returns a list of card entity order by name in the ascending order (case insensitive)	There are 5 cards entities in the database, with the names "sam", "jack", "peter", "adam", "Zerg"	val cards = cardEntityDao.getAllCardsOrderByName();	return a list of cards entities in the order of "adam", "jack", "peter", "sam", "Zerg"	return a list of cards entities in the order of "adam", "jack", "peter", "sam", "Zerg"	PASS	
4	InsertCardListCrossReference: Insert a cross reference entity for the association between card entity and list entity	There is a cardListCrossRef object with cardId and listId	cardListCrossRefDao.insert(cardListCrossRef);	The cardListCrossRef object is inserted into the database	The cardListCrossRef object is inserted into the database	PASS	
5	LetListsWithCards: Get all lists with cards in each list	There are two list entities object, named "first list" and "second list" in the database. The relationship between list entity and cards entity is defined in CardListCrossReference table. "First list" has two card entities with cardId=50 and cardId=100. The "Second list" has no cards belong to it.	val results: List<ListWithCardsEntity> = listEntityDao.getListWithCards()	Return "First list" with two cards entities with cardId=50 and cardId=100. And "Second list" with no card entity.	Return "First list" with two cards entities with cardId=50 and cardId=100. And "Second list" with no card entity.	PASS	