UCID		Name	
Measure	Weight	Evaluation	Score
Design	35%	Design is evaluated by the UML submission.	0.00%
Code	35%	Code is evaluated by running the program and looking at the code.	0.00%
Testing	30%	Testing is evaluated by running the tests, reviewing the tests, and calculating test coverage.	0.00%
		Total	0.00%
		The grade will be entered in D2L as the following value out of 100 (rounded to nearest 10th):	0.0

Design (30%)	***Important: If the 'Readability' requirements are not all met, the other measures will not be evaluated and you will receive zero for Design. Readability contributes no points to the total. For Yes/No options, the project must completely meet the requirement to receive a Yes.***	Category	Rating	Options
	The UML diagram details the program design, showing class relationships, methods, class and instance variables, parameter types, return value types, access modifiers (public, private, etc), other modifiers (i.e., static, final and abstract) and cardinalities. The class containing main() does not need to be shown if it contains no other methods. The database can be depicted as a single rectangle.	Readability		Yes/No
	The diagram is consistent (e.g., relationships are supported by data members) and appears to be an accurate representation of the code.	Readability		Yes/No
	Diagram is legible - lines are tidy, text can be read, and fits on the page(s) neatly. The file is a PDF.	Readability		Yes/No
	Each class has a clear, distinct purpose (encapsulation)	OOP		90-100% (8), 70-89% (6), 50-69% (4), <50% (0)
	Inheritance and/or interfaces are used in the design, and used appropriately	OOP		
	Aggregation and/or composition is used in the design, and is used appropriately	OOP		
	Pattern (Singleton, Builder, Observer, Strategy, or M-V-C) and/or generics is used in design, and is used appropriately	OOP		
	Method overloading is used (polymorphism), and is used appropriately	OOP		Yes (2) /No (0)
	Single Responsibility Principle is followed throughout (if Singleton pattern is used appropriately, violation of SRP is permitted in this instance)	OOP		Yes (4) / No (0)
	Open-Closed Principle is followed throughout	OOP		
	Liskov Substitution Principle is followed throughout	OOP		
	Interface Substitution Principle is followed throughout	OOP		
	Dependency Inversion Principle is followed throughout	OOP		
	Public methods are not assumed to receive correct imput and throw exceptions accordingly	Java		90-100% (4), 70-89% (3), 50-69% (2), <50% (0)
	Class names are appropriate (singular, indicate purpose)	Java		
	Method names indicate expected behavior and follow naming conventions	Java		
	Variable types (including data structures) are appropriate for purpose	Java		
Grader Remarks		Total marks		
		Percentage		

Code (30%)	***Important: If the code does not meet all the requirements in the 'Compatibility' category, the other measures will not be evaluated and you will receive a zero for Code. Compatibility contributes no points toward the total. The code will be run with different test data than was provided as an example. All evaluations will be based on how the code reacts with our testing data. For Yes/No options, the project must completely meet the requirement to receive a Yes.***	Category	Rating	Options	Points	Out of
	The code is provided as a zipfile, with the UCID as the name of the file (e.g., UCID 12345678 is 12345678.zip).	Compatibility		Yes/No	0	0
	The edu.ucalgary.oop package is used, and all code is located in the directory: src/edu/ucalgary/oop	Compatibility		Yes/No	0	0
	All input data files are located in the directory: data	Compatibility		Yes/No	0	0
	The code (excluding tests) can be compiled from the command-line using Java 21, using only the following external libraries: postgresql.	Compatibility		Yes/No	0	0
	The code can be run from the command-line, using only the following external libraries: postgresql.	Compatibility		Yes/No	0	0
	The code connects to the PostgreSQL database using the username 'oop' and the password 'ucalgary'.	Compatibility		Yes/No	0	0
	The database uses the design which was provided, without modifications.	Compatibility		Yes/No	0	0
	The application can be run as edu.ucalgary.oop.Main (i.e., main() is located in a class named Main.java)	Compatibility		Yes/No	0	0
	Documentation can be generated for all (code, not test) files using javadoc.	Dev docs		Yes (4) / No (0)	0	4
	For the following 'Dev docs' evaluations, one class is randomly selected. Selected class:					
	File contains appropriate header information (author, version, date).	Dev docs		Yes (1) / No (0)	0	1
	Every method is adequately documented with @param, @return and information about the purpose.	Dev docs		Yes (5) / No (0)	0	5
	The user interface to the application allows relief workers to enter information about disaster victims, including their relationships, medical records, and location. it allows inquiries to be logged and supplies to be allocated to an individual or location. Existing data can be modified and viewed.	UI Feature		Yes (full points) No (0 points)	0	10
	The user interface enforces data type validity (data which is outside of range or incorrect type is not inserted)	UI Feature			0	3
	The user interface allows user to correct errors (e.g., typing B instead of a number 1-4) and lets them know what appropriate values are	UI Feature			0	3
	When the program cannot recover from an error, it appends to the logfile errorlog.txt with timestamp and exception information	UI Feature			0	4
	When the program cannot recover from an error, it gives the user a message and exits cleanly	UI Feature			0	2
	The user interface does not require unnecessary typing	UI Feature			0	5
	Obsolete data (such as expired water) is deleted at startup	DB Feature			0	2
	The program is populated with data from the database at startup	DB Feature			0	8
	The user's additions and modifications are reflected in the database	DB Feature			0	10
	The gender options which are provided are man, woman, and non-binary person	Gender Feature			0	1
	Command-line language selection is supported as described in feature (no need to specify extension, etc)	Lang Feature			0	1
	The program falls back to en-CA in case of error	Lang Feature			0	2
	The provided language file is in the correct XML format (if this is not done, the next three criteria will automatically be marked No)	Lang Feature			0	1
	When a modified language file is used, input prompts are translated	Lang Feature			0	5
	When a modified language file is used, error messages are translated	Lang Feature			0	5
	When a modified language file is used, variable parts of the text are translated (excluding data pulled from the database)	Lang Feature			0	5
	Each person can be a member of only one family group	Fam Feature			0	1
	Each family group can have one or more members	Fam Feature			0	1
	Personal belonging with description exists, cannot be associated with location	Inv. Feature			0	1
	Cot inventory has a room (format: number) and grid (format letter + number) attribute; incorrect data cannot be inserted	Inv. Feature			0	1
	Water inventory is used (removed from inventory) one day after allocation to person	Inv. Feature			0	1
	Blanket type inventory exists	Inv. Feature			0	1
	Inventory can be allocated to disaster victims or locations; when inventory is allocated to a person it is removed from location	Inv. Feature			0	3
	Only inventory at the same location as a disaster victim can be allocated to that person	Inv. Feature			0	1
	Disaster victims can log inquiries	Inquiry Feature			0	2
	No redundant information about the same individual is stored	Inquiry Feature			0	3
	The program has an option for the user to stop running the program. It is possible to perform all tasks without needing to exit the program (there is navigation between menu levels).	Functional			0	3
	The program does not crash during evaluation.	Functional			0	5
Grader Remarks		**Total marks**			0	100
		Percentage			0.00%	

Testing (25%)	***Important: If the tests do not meet all the requirements in the 'Compatibility' category, the other measures will not be evaluated and the you will receive a zero for Testing. Compatibility contributes no points toward the total. For Yes/No options, the project must completely meet the requirement to receive a Yes.***	Category	Rating	Options	Points	Out of
	The code is provided as a zipfile, with the UCID as the name of the file (e.g., UCID 12345678 is 12345678.zip).	Compatibility		Yes/No	0	0
	The edu.ucalgary.oop package is used, and tests are in test/edu/ucalgary/oop	Compatibility		Yes/No	0	0
	The tests can be compiled from the command-line, using only the following external libraries: JUnit 4, hamcrest.	Compatibility		Yes/No	0	0
	The tests can be run from the command-line, using only the following external libraries: JUnit 4, hamcrest.	Compatibility		Yes/No	0	0
	There are no tests which involve connecting to the database or user interface.	Compatibility		Yes/No	0	0
	All test files include 'Test' in the name. Only test files include 'Test' in the name.	Compatibility		Yes/No	0	0
	Code coverage is good (a lower percentage is allowed for top marks to account for the user interface and database connection not being tested)	Coverage		65-100% (15),50-64% (10), 40-49% (5),<49% (0)	0	15
	One class is examined manually at random to evaluate the next test coverage assessment. Name of evaluated class:					
	Edge cases and boundary conditions have been considered.	Coverage		All (15), 1-3 missing (10), 4-7 missing (5), > 7 missing (0)	0	15
	All tests pass as provided.	Pass		Yes (5) /No (0)	0	5
	Tests do not have a dependency on existing external files (if files are needed, they are written by tests and cleaned by tests).	Pass		Yes (5) /No (0)	0	5
	Three tests in different test files will be randomly spot-checked for the following assessments. Names of evaluated tests:					
	Tests use AAA structure.	Design		All 3 tests meet criteria (3), two tests meet criteria (2), one test meets criteria (1), no tests meet criteria (0)	0	3
	Tests have meaningful names which convey information about testing scenario.	Design			0	3
	Tests follow Java naming convention (begin with 'test').	Design			0	3
	Tests have useful error messages (except in cases where exception is expected, this point is automatically given).	Design			0	3
	Tests only test a single scenario (usually only one assert statement).	Design			0	3
	Tests are self-contained (setup information may be in @Before).	Design			0	3
Grader Remarks		**Total marks**			0	58
		Percentage			0.00%	