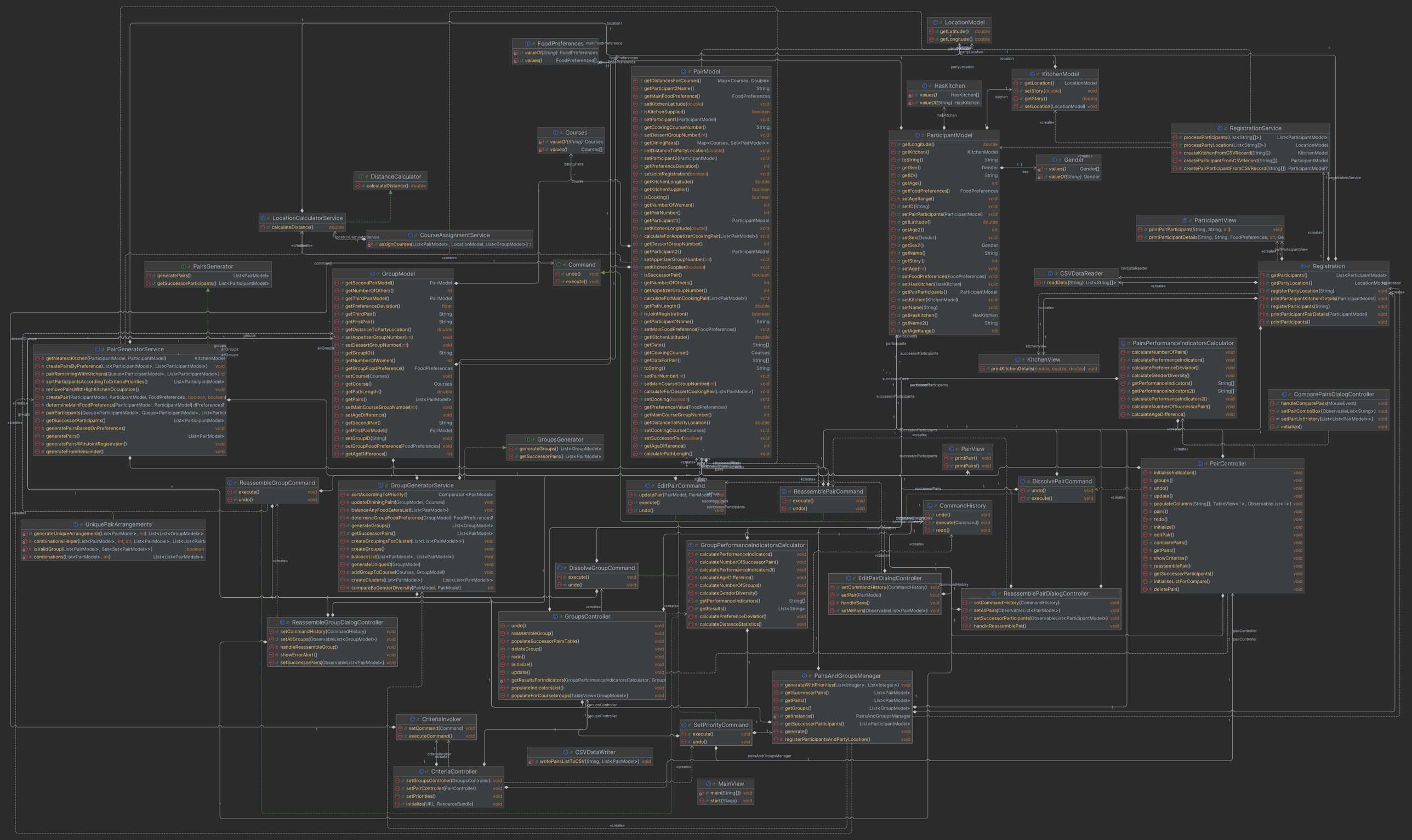
FoodPreferences
valueOf(String) FoodPreferences
values() FoodPreferences[]
LocationModel
getLatitude() double
getLongitude() double
PairModel
getDistancesForCourses() Map<Courses, Double>
getParticipant2Name() String
getMainFoodPreference() FoodPreferences
setKitchenLatitude(double) void
isKitchenSupplier() boolean
setParticipant1(ParticipantModel) void
getCookingCourseNumber() String
setDessertGroupNumber(int) void
getDiningPairs() Map<Courses, Set<PairModel>>
setDistanceToPartyLocation(double) void
setParticipant2(ParticipantModel) void
getPreferenceDeviation() int
setJointRegistration(boolean) void
getKitchenLongitude() double
getKitchenSupplier() boolean
isCooking() boolean
getNumberOfWomen() int
getPairNumber() int
getParticipant1() ParticipantModel
setKitchenLongitude(double) void
calculateForAppetizerCookingPair(List<PairModel>) void
getDessertGroupNumber() int
getParticipant2() ParticipantModel
setAppetizerGroupNumber(int) void
setKitchenSupplier(boolean) void
isSuccessorPair() boolean
getNumberOfOthers() int
getAppetizerGroupNumber() int
calculateForMainCookingPair(List<PairModel>) void
getPathLength() double
isJointRegistration() boolean
getParticipant1Name() String
setMainFoodPreference(FoodPreferences) void
getKitchenLatitude() double
getData() String[]
getCookingCourse() Courses
getDataForPair() String[]
toString() String
setPairNumber(int) void
setMainCourseGroupNumber(int) void
calculateForDessertCookingPair(List<PairModel>) void
setCooking(boolean) void
getPreferenceValue(FoodPreferences) int
getMainCourseGroupNumber() int
getDistanceToPartyLocation() double
setCookingCourse(Courses) void
setSuccessorPair(boolean) void
getAgeDifference() int
calculatePathLength() void
Courses
valueOf(String) Courses
values() Courses[]
DistanceCalculator
calculateDistance() double
LocationCalculatorService
calculateDistance() double
CourseAssignmentService
assignCourses(List<PairModel>, LocationModel, List<GroupModel>)
PairsGenerator
generatePairs() List<PairModel>
getSuccessorParticipants() List<ParticipantModel>
Command
undo() void
execute() void
GroupModel
getSecondPairModel() PairModel
getNumberOfOthers() int
getThirdPairModel() PairModel
getPreferenceDeviation() float
getThirdPair() String
getFirstPair() String
getDistanceToPartyLocation() double
getAppetizerGroupNumber(int) void
setDessertGroupNumber(int) void
getGroupID() String
getNumberOfWomen() int
getGroupFoodPreference() FoodPreferences
setCourse(Courses) void
getCourse() Courses
getPathLength() double
getPairs() List<PairModel>
setMainCourseGroupNumber(int) void
setAgeDifference() void
getSecondPair() String
getFirstPairModel() PairModel
setGroupID(String) void
setGroupFoodPreference(FoodPreferences) void
getAgeDifference() int
PairGeneratorService
getNearestKitchen(ParticipantModel, ParticipantModel) KitchenModel
createPairsByPreference(List<ParticipantModel>, List<ParticipantModel>) void
pairRemainingWithKitchens(Queue<ParticipantModel>, List<ParticipantModel>) id
sortParticipantsAccordingToCriteriaPriorities() List<ParticipantModel>
removePairsWithHighKitchenOccupation() void
createPair(ParticipantModel, ParticipantModel, FoodPreferences, boolean, boolean)
determineMainFoodPreference(ParticipantModel, ParticipantModel) dPreference
pairParticipants(Queue<ParticipantModel>, Queue<ParticipantModel>, List<Partici
getSuccessorParticipants() List<ParticipantModel>
generatePairsBasedOnPreference()
generatePairs() List<PairModel>
generatePairsWithJointRegistration()
generateFromRemainder()
GroupsGenerator
generateGroups() List<GroupModel>
getSuccessorPairs() List<PairModel>
GroupGeneratorService
sortAccordingToPriority() Comparator<PairModel>
updateDiningPairs(GroupModel, Courses) void
balanceAnyFoodEatersList(List<PairModel>) void
determineGroupFoodPreference(GroupModel) FoodPreference
generateGroups() List<GroupModel>
getSuccessorPairs() List<PairModel>
createGroupingsForCluster(List<List<PairModel>>) void
createGroups() void
balanceList(List<PairModel>, List<PairModel>) void
generateUniqueID(GroupModel) void
addGroupToCourse(Courses, GroupModel) void
createClusters(List<PairModel>) List<List<PairModel>>
compareByGenderDiversity(PairModel, PairModel) int
UniquePairArrangements
generateUniquePairArrangements(List<PairModel>, int) List<List<GroupModel>>
combinationsHelper(List<PairModel>, int, int, List<PairModel>, List<List<PairM
isValidGroup(List<PairModel>, Set<Set<PairModel>>) boolean
combinations(List<PairModel>, int) List<List<PairModel>>
ReassembleGroupCommand
execute() void
undo() void
EditPairCommand
updatePair(PairModel, PairModel) void
execute() void
undo() void
ReassemblePairCommand
execute() void
undo() void
DissolveGroupCommand
execute() void
undo() void
GroupPerformanceIndicatorsCalculator
calculatePerformanceIndicators() void
calculateNumberOfSuccessorPairs() void
calculatePerformanceIndicators2() void
calculateAgeDifference() void
calculateNumberOfGroups() void
calculateGenderDiversity() void
getPerformanceIndicators() String[]
getResults() List<String>
calculatePreferenceDeviation() void
calculateDistanceStatistics() void
GroupsController
undo() void
reassembleGroup() void
populateSuccessorPairsTable() void
deleteGroup() void
redo() void
initialize() void
update() void
getResultsForIndicators(GroupPerformanceIndicatorsCalculator, Group
populateIndicatorsList() void
populateForCourseGroups(TableView<GroupModel>) void
ReassembleGroupDialogController
setCommandHistory(CommandHistory) void
setAllGroups(ObservableList<GroupModel>) void
handleReassembleGroup() void
showErrorAlert() void
setSuccessorPairs(ObservableList<PairModel>) void
CriteriaInvoker
setCommand(Command) void
executeCommand() void
CriteriaController
setGroupsController(GroupsController) void
setPairController(PairController) void
setPriorities() void
initialize(URL, ResourceBundle) void
CSVDataWriter
writePairsListToCSV(String, List<PairModel>) void
SetPriorityCommand
execute() void
undo() void
PairsAndGroupsManager
generateWithPriorities(List<Integer>, List<Integer>) void
getSuccessorPairs() List<PairModel>
getPairs() List<PairModel>
getGroups() List<GroupModel>
getInstance() PairsAndGroupsManager
getSuccessorParticipants() List<ParticipantModel>
generate() void
registerParticipantsAndPartyLocation() void
MainView
main(String[]) void
start(Stage) void
HasKitchen
values() HasKitchen[]
valueOf(String) HasKitchen
KitchenModel
getLocation() LocationModel
setStory(double) void
getStory() double
setLocation(LocationModel) void
ParticipantModel
getLongitude() double
getKitchen() KitchenModel
toString() String
getSex() Gender
getID() String
getAge() int
getFoodPreferences() FoodPreferences
setAgeRange() void
setID(String) void
setPairParticipants(ParticipantModel) void
getLatitude() double
getAge2() int
setSex(Gender) void
getSex2() Gender
getName() String
getStory() int
setAge(int) void
setFoodPreferences(FoodPreferences) void
setHasKitchen(HasKitchen) void
getPairParticipants() ParticipantModel
setKitchen(KitchenModel) void
setName(String) void
getHasKitchen() HasKitchen
getName2() String
getAgeRange() int
Gender
values() Gender[]
valueOf(String) Gender
RegistrationService
processParticipants(List<String[]>) List<ParticipantModel>
processPartyLocation(List<String[]>) LocationModel
createKitchenFromCSVRecord(String[]) KitchenModel
createParticipantFromCSVRecord(String[]) ParticipantModel
createPairParticipantFromCSVRecord(String[]) ParticipantModel?
ParticipantView
printPairParticipant(String, String, int) void
printPairParticipantDetails(String, String, FoodPreferences, int, Ge
CSVDataReader
readData(String) List<String[]>
Registration
getParticipants() List<ParticipantModel>
getPartyLocation() LocationModel
registerPartyLocation(String) void
printParticipantKitchenDetails(ParticipantModel) void
registerParticipants(String) void
printPairParticipantDetails(ParticipantModel) void
printParticipants() void
KitchenView
printKitchenDetails(double, double, double) void
PairsPerformanceIndicatorsCalculator
calculateNumberOfPairs() void
calculatePerformanceIndicators() void
calculatePreferenceDeviation() void
calculateGenderDiversity() void
getPerformanceIndicators() String[]
getPerformanceIndicators2() String[]
calculatePerformanceIndicators2() void
calculateNumberOfSuccessorPairs() void
calculateAgeDifference() void
ComparePairsDialogController
handleComparePairs(MouseEvent) void
setPairComboBox(ObservableList<String>) void
setPairListHistory(List<List<PairModel>>) void
initialize() void
PairView
printPair() void
printPairs() void
DissolvePairCommand
undo() void
execute() void
CommandHistory
undo() void
execute(Command) void
redo() void
EditPairDialogController
setCommandHistory(CommandHistory) void
setPair(PairModel) void
handleSave() void
setAllPairs(ObservableList<PairModel>) void
ReassemblePairDialogController
setCommandHistory(CommandHistory) void
setAllPairs(ObservableList<PairModel>) void
setSuccessorParticipants(ObservableList<ParticipantModel>) void
handleReassemblePair() void
PairController
initialiseIndicators() void
groups() void
undo() void
update() void
populateColumns(String[], TableView<?>, ObservableList<?>) void
pairs() void
redo() void
initialize() void
editPair() void
comparePairs() void
getPairs() void
showCriterias() void
reassemblePair() void
getSuccessorParticipants() void
initialiseListForCompare() void
deletePair() void