

## Package FXML

FileChooserClass	
f	desktop Desktop
m	onClick(Stage) File
m	saveFile(File, int, String) File
m	openFile(File) void

MainUI	
f	stage Stage
f	model DataModel
f	sm SerializerManager
m	MainUI()
m	main(String[]) void
m	start(Stage) void

ArrangeInterviewsController	
f	model DataModel
m	ArrangeInterviewsController(DataModel)
m	arrangeButton(ActionEvent) void
m	hrAssign(ActionEvent, int) void
m	moveHRMenu(ActionEvent) void

DataModel	
f	jobportal JobPortal
f	name String
f	password String
f	id int
f	user User
f	cv String
f	cl String
m	DataModel(JobPortal)

SignupController	
f	type String
f	typeSelection ChoiceBox<String>
f	signupName TextField
f	signupPassword TextField
m	signupSubmit(ActionEvent) void
m	setUpModel() void

LoginController	
f	model DataModel
f	userIdInput TextField
f	passwordInput TextField
f	userid String
f	password String
f	id int
m	LoginController(DataModel)

ViewJobsController	
f	tableView TableView<JobPosting>
f	postingId TableColumn<JobPosting, String>
f	postingCompany TableColumn<JobPosting, String>
f	postingName TableColumn<JobPosting, String>
f	postingOpenDate TableColumn<JobPosting, String>
f	postingCloseDate TableColumn<JobPosting, String>
f	postingRequirements TableColumn<JobPosting, String>
f	filterFieldTag TextField
f	filterFieldName TextField
f	filterFieldCompany TextField
f	filterFieldRequirements TextField
f	listings ArrayList<JobPosting>
f	model DataModel
f	filteredListing FilteredList<JobPosting>
f	sortedListing SortedList<JobPosting>
m	ViewJobsController(DataModel)

m	userExists(int) boolean
m	correctPassword(int, String) boolean
m	setUser(User) void
m	setNewApplicant() void
m	setNewHRCor(Company) void
m	setNewInterviewer(Company) void
m	setName(String) void
m	setPassword(String) void
m	setCV(String) void
m	setCL(String) void
m	setId() void
m	addToUserlist() void
m	getUserType() String
m	getUserApplicant() Applicant
m	getUserHRCoord() HRCoord
m	getUserInterviewer() Interviewer

ApplicantMenuController	
f	model DataModel
f	ap AnchorPane
f	applicantID Label
m	ApplicantMenuController(DataModel)
m	initialize() void
m	viewJobPostings(ActionEvent) void
m	replaceDocuments() void
m	viewDocuments() void
m	applyJobs() void
m	viewStatus(ActionEvent) void
m	viewHistory(ActionEvent) void
m	withdraw() void
m	rateInterviewer(ActionEvent) void
m	applicantLogout(ActionEvent) void
m	verifyAdd(String) JobPosting

m	isNumeric(String) boolean
m	loginUserSuccess() boolean
m	loginButton(ActionEvent) void
m	signupButton(ActionEvent) void
m	moveToApplicantMenu(ActionEvent) void
m	moveHRCorMenu(ActionEvent) void
m	moveInterviewerMenu(ActionEvent) void

HRCoordMenuController	
f	ap AnchorPane
f	addJobButton Button
f	model DataModel
m	HRCoordMenuController(DataModel)
m	viewCompanyJobs() void
m	addNewJob(ActionEvent) void
m	initialize() void
m	setDisabled() void
m	arrangeInterviews(ActionEvent) void
m	viewCompanyJobs(ActionEvent) void
m	viewApplicantFiles() void
m	addNewInterviewType() void
m	hrLogout(ActionEvent) void

m	initialize() void
m	createTable() void
m	setFilterFieldTag() void
m	setFilterFieldName() void
m	setFilterFieldCompany() void
m	setFilterFieldRequirements() void
m	submitButton Button
m	backButton(ActionEvent) void

InterviewerMenu	
f	model DataModel
f	ap AnchorPane
m	InterviewerMenu(DataModel)
m	interviewerSignup(ActionEvent) void
m	recommendApplicants() void
m	viewDocuments() void
m	viewProfile(ActionEvent) void
m	interviewerLogout(ActionEvent) void

SetupController	
f	model DataModel
f	givenUserId Label
f	ap AnchorPane
m	SetupController(DataModel)
m	onClick(ActionEvent) void
m	initialize() void
m	applicantSignup(ActionEvent) void
m	moveToApplicantMenu(ActionEvent) void
m	createUser() void

ApplicantRatingController	
f	model DataModel
f	interviewerId TextField
f	jobName TextField
f	ratingSelect ChoiceBox<Integer>
m	ApplicantRatingController(DataModel)
m	initialize() void
m	onSubmit() void
m	setRatingList() void
m	backButton(ActionEvent) void

ViewApplicantHistory	
f	historyView TableView<JobApplication>
f	historyName TableColumn<JobApplication, String>
f	historyCompany TableColumn<JobApplication, String>
f	historyApplied TableColumn<JobApplication, String>
f	dateCreated Label
f	model DataModel
m	ViewApplicantHistory(DataModel)
m	initialize() void
m	createTable() void
m	backButton(ActionEvent) void

ArrangeInterviewMenuController	
f	model DataModel
f	interviewerSelected List<String>
f	id int
f	interviewerList ListView
m	ArrangeInterviewMenuController(DataModel, int)
m	initialize() void
m	backButton(ActionEvent) void
m	onSubmit(ActionEvent) void
m	setInterviewerList() void
m	setApplicantList() void

DisplayInvitedInterviews	
f	interviews TableView<Interview>
f	interviewee TableColumn<Interview, String>
f	type TableColumn<Interview, String>
f	jobTitle TableColumn<Interview, String>
f	date TableColumn<Interview, String>
f	model DataModel
m	DisplayInvitedInterviews(DataModel)
m	createTable() void
m	backButton(ActionEvent) void
m	onSubmit() void

ViewInterviewerProfile	
f	interviewerName TextField
f	interviewerId TextField
f	interviewerCompany TextField
f	interviewerRatings TextField
f	model DataModel
m	ViewInterviewerProfile(DataModel)
m	initialized() void
m	setText() void

ViewApplicantStatus	
f	statusView TableView<Interview>
f	statusName TableColumn<Interview, String>
f	currentStatusTableColumn<Interview, String>
f	model DataModel
m	ViewApplicantStatus(DataModel)
m	initialize() void
m	createTable() void
m	backButton(ActionEvent) void

AddJobsController	
f	model DataModel
f	addJobsTitle TextField
f	addJobsRequirements TextField
f	tagField TextField
f	closingDate DatePicker
f	numPosition TextField
f	interviewTypes ArrayList<ArrayList<Object>>
m	AddJobsController(DataModel, ArrayList<ArrayList<Object>>)
m	selectInterviewTypes(ActionEvent) void
m	backButton(ActionEvent) void
m	addJobsSubmit(ActionEvent) void