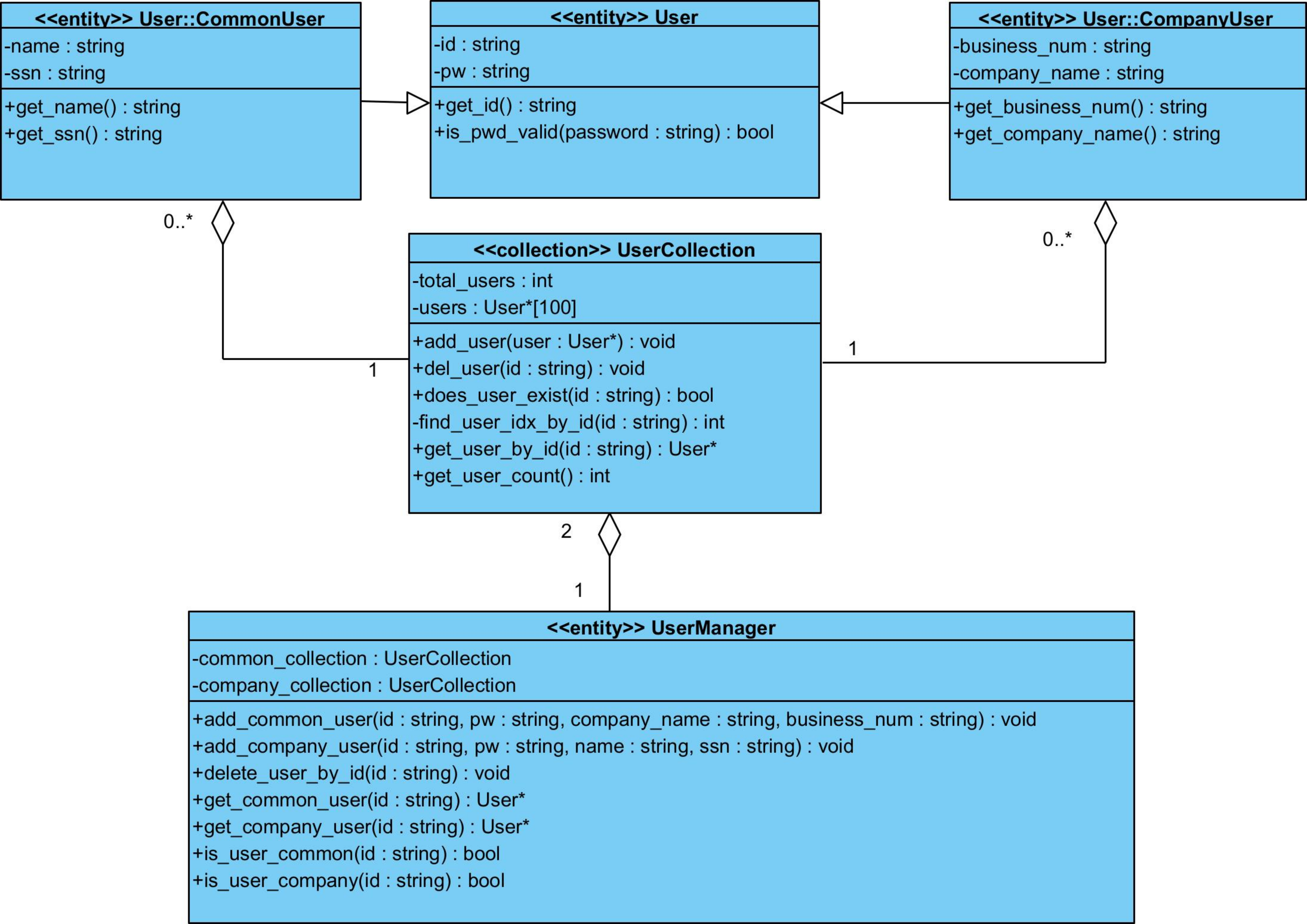

<<entity>> User::CommonUser
-name : string
-ssn : string
+get_name() : string
+get_ssn() : string
<<entity>> User
-id : string
-pw : string
+get_id() : string
+is_pwd_valid(password : string) : bool
<<entity>> User::CompanyUser
-business_num : string
-company_name : string
+get_business_num() : string
+get_company_name() : string
0..*
0..*
<<collection>> UserCollection
-total_users : int
-users : User*[100]
+add_user(user : User*) : void
+del_user(id : string) : void
+does_user_exist(id : string) : bool
-find_user_idx_by_id(id : string) : int
+get_user_by_id(id : string) : User*
+get_user_count() : int
1
1
2
1
<<entity>> UserManager
-common_collection : UserCollection
-company_collection : UserCollection
+add_common_user(id : string, pw : string, company_name : string, business_num : string) : void
+add_company_user(id : string, pw : string, name : string, ssn : string) : void
+delete_user_by_id(id : string) : void
+get_common_user(id : string) : User*
+get_company_user(id : string) : User*
+is_user_common(id : string) : bool
+is_user_company(id : string) : bool

채용 관리 시스템

<<boundary>> User Interface::CompanyNameSearchUI

-company_names : string[100]
-control : CompanyNameSearch*
-out_stream : ofstream*

+start_interface(in_fp : ifstream*, out_fp : ofstream*) : void
+write_employment(company_name : string, business_name : string, work_type : string, total_applicants : int, dead_line : string) : void

<<control>> Control::CompanyNameSearch

-collection : EmploymentCollection*
-ui : CompanyNameSearchUI*

+search_company_name(company_name : string) : void

<<boundary>> User Interface::ShowEmploymentListUI

-control : ShowEmploymentList*
-out_stream : ofstream*

+start_interface(logged_in_user_id : string, out_fp : ofstream*) : void
+write_employment(work_type : string, total_applicants : int, dead_line_string) : void

<<control>> ShowEmploymentList

-collection : EmploymentCollection*
-ui : ShowEmploymentListUi*
-manager : UserManager*

+show_employment_info(logged_in_user_id : string) : void

<<boundary>> User Interface::EmploymentAddUI

-control : EmploymentAdd

+start_interface(user_id : string, in_fp : ifstream*, out_fp : ofstream*) : void

<<control>> EmploymentAdd

-collection : EmploymentCollection*
-manager : UserManager*

+add_employment(user_id : string, work_type : string, max_applicants : int, deadline : string) : void

채용 지원 시스템

<<Boundary>>ApplicationCheckUI

-control : ApplicationCheck*
-out_stream : ofstream*

+start_interface(logged_in_user_id : string, out_fp : ofstream*) : void
+write_application(company_name : string, business_num : string, work_type : string, max_applicants : int, dead_line : string) : void

<<Control>>ApplicationCheck

-collection : ApplicationCollection*
-ui : ApplicationCheckUI

+show_applications_info(logged_in_user_id : string) : void

<<Boundary>>CancelApplicationUI

-control : CancelApplication

+start_interface(logged_in_user_id : string, in_fp : ifstream*, out_fp : ofstream*) : void

<<Control>>CancelApplication

-collection : ApplicationCollection*

+cancel_application(logged_in_user_id : string, business_num : string) : Application*

<<Boundary>>NowApplicationUi

-control : NowApplication

+start_interface(logged_in_user_id : string, in_fp : ifstream*, out_fp : ofstream*) : void

<<Control>> NowApplication

-application_collection : ApplicationCollection*
-employment_collection : EmploymentCollection*

+add_new_application(logged_in_user_id : string, business_num : string) : Application*

통계 시스템

<<boundary>> User Interface::EmploymentStatUI

-control : ShowEmploymentStat*
-out_stream : ofstream*

+start_interface(logged_in_user_id : string, out_fp : ofstream*) : void
+output_statistics(work_keyword : string, total : int) : void

<<boundary>> User Interface::ApplicationStatUI

-control : ShowApplicationStat*
-out_stream : ofstream*

+start_interface(logged_int_user_id : string, out_fp : ofstream*) : void
+output_statistics(work_keyword : string, total : int) : void

<<control>> Control::ShowEmploymentStat

-application_collection : ApplicationCollection*
-employment_collection : EmploymentCollection*
-ui : EmploymentStatUI
-manager : UserManager*

+run_all_statistics(id : string) : void
-run_statistics(keyword : string, id : string) : int

<<control>> Control::ShowApplicationStat

-application_collection : ApplicationCollection*
-employment_collection : EmploymentCollection*
-ui : ApplicationStatUI*
-manager : UserManager*

+run_all_statistics(id : string) : void
-run_statistics(keyword : string, id : string) : int

회원 관리 시스템

<<boundary>> User Interface::LogoutUI

-control : LogoutManager

+start_interface(logged_in_user_id : string, out_fp : ofstream*) : void

<<boundary>> User Interface::LoginUI

-control : LoginManager

+start_interface(in_fp : ifstream*, out_fp : ofstream*) : string

<<control>> Control::LogoutManager

-manager : UserManager*

+logout(id : string) : void

<<control>> Control::LoginManager

-manager : UserManager*

+login(id : string, pw : string) : User*

<<boundary>> User Interface::AddCommonUserUI

-control : AddNewCommonUser

+start_interface(in_fp : ifstream*, out_fp : ofstream*) : void

<<boundary>> User Interface::AddCompanyUser

-control : AddNewCompanyUser

+start_interface(in_fp : ifstream*, out_fp : ofstream*) : void

<<control>> Control::AddNewCommonUser

-manager : UserManager*

+add_user(id : string, pw : string, name : string, ssn : string) : void

<<control>> Control::AddNewCompanyUser

-manger : UserManager*

+add_user(id : string, pw : string, company_name : string, business_num : string) : void

<<boundary>> User Interface::UserWithdrawalUI

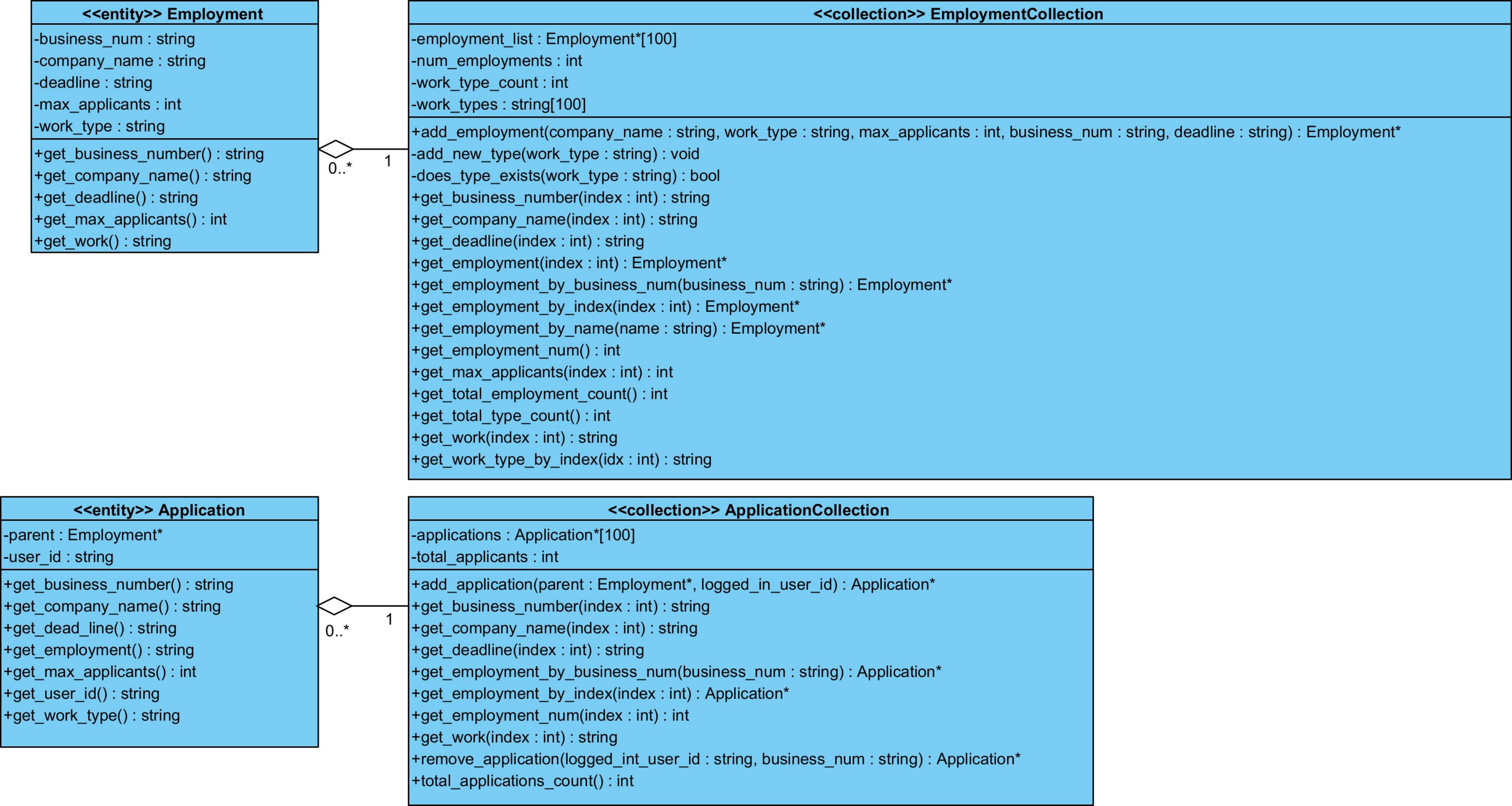
-control : UserWithdrawal

+start_interface(logged_in_user_id : string, out_fp : ofstream*) : void

<<control>>Control::UserWithdrawal

-manager : UserManager*

+withdrawal_user(logged_in_user_id : string) : void


<<entity>> Employment
-business_num : string
-company_name : string
-deadline : string
-max_applicants : int
-work_type : string
+get_business_number() : string
+get_company_name() : string
+get_deadline() : string
+get_max_applicants() : int
+get_work() : string
0..*
1
<<collection>> EmploymentCollection
-employment_list : Employment*[100]
-num_employments : int
-work_type_count : int
-work_types : string[100]
+add_employment(company_name : string, work_type : string, max_applicants : int, business_num : string, deadline : string) : Employment*
-add_new_type(work_type : string) : void
-does_type_exists(work_type : string) : bool
+get_business_number(index : int) : string
+get_company_name(index : int) : string
+get_deadline(index : int) : string
+get_employment(index : int) : Employment*
+get_employment_by_business_num(business_num : string) : Employment*
+get_employment_by_index(index : int) : Employment*
+get_employment_by_name(name : string) : Employment*
+get_employment_num() : int
+get_max_applicants(index : int) : int
+get_total_employment_count() : int
+get_total_type_count() : int
+get_work(index : int) : string
+get_work_type_by_index(idx : int) : string
<<entity>> Application
-parent : Employment*
-user_id : string
+get_business_number() : string
+get_company_name() : string
+get_dead_line() : string
+get_employment() : string
+get_max_applicants() : int
+get_user_id() : string
+get_work_type() : string
0..*
1
<<collection>> ApplicationCollection
-applications : Application*[100]
-total_applicants : int
+add_application(parent : Employment*, logged_in_user_id) : Application*
+get_business_number(index : int) : string
+get_company_name(index : int) : string
+get_deadline(index : int) : string
+get_employment_by_business_num(business_num : string) : Application*
+get_employment_by_index(index : int) : Application*
+get_employment_num(index : int) : int
+get_work(index : int) : string
+remove_application(logged_int_user_id : string, business_num : string) : Application*
+total_applications_count() : int