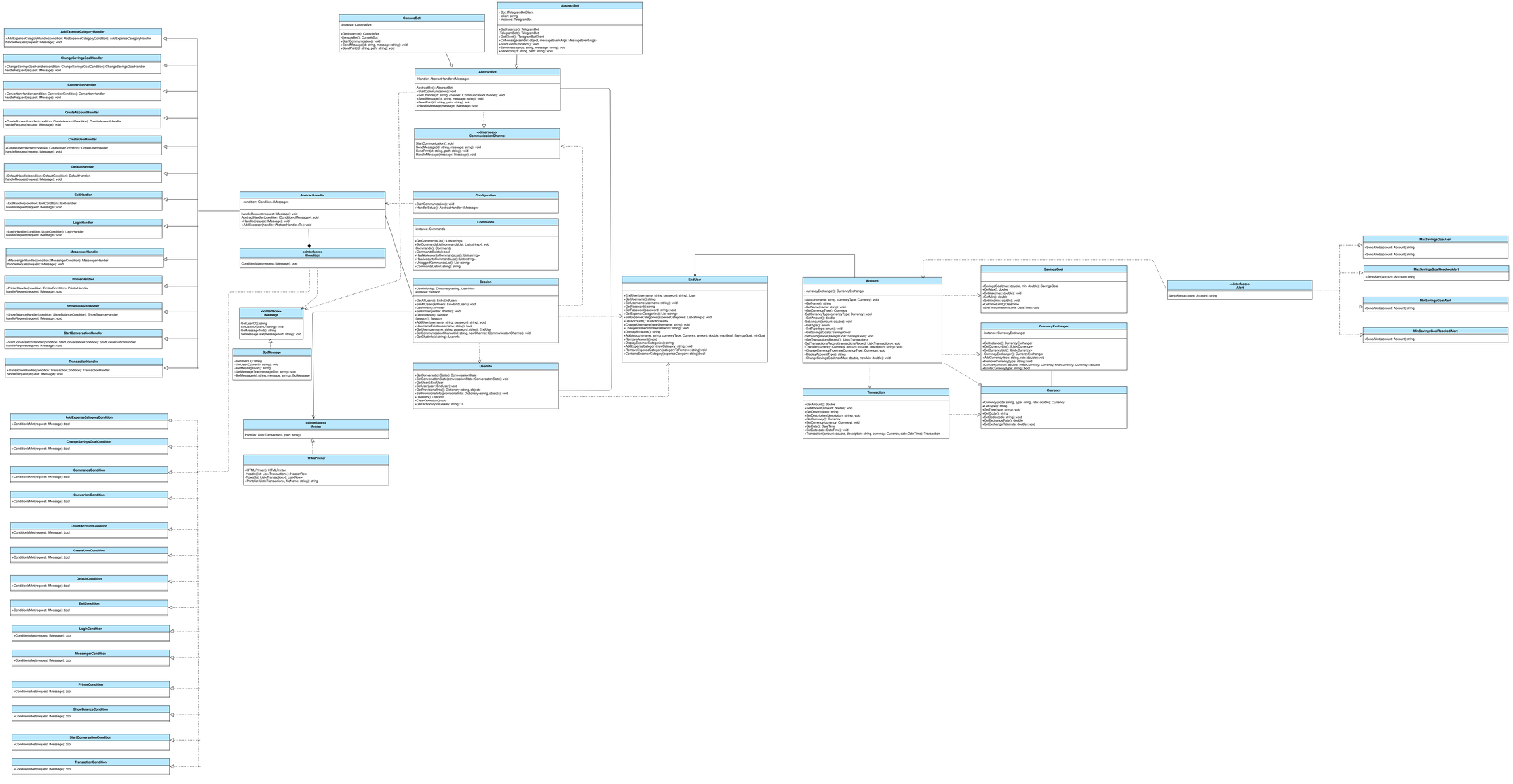
AddExpenseCategoryHandler
+AddExpenseCategoryHandler(condition: AddExpenseCategoryCondition): AddExpenseCategoryHandler
handleRequest(request: IMessage): void
ChangeSavingsGoalHandler
+ChangeSavingsGoalHandler(condition: ChangeSavingsGoalCondition): ChangeSavingsGoalHandler
handleRequest(request: IMessage): void
ConvertionHandler
+ConvertionHandler(condition: ConvertionCondition): ConvertionHandler
handleRequest(request: IMessage): void
CreateAccountHandler
+CreateAccountHandler(condition: CreateAccountCondition): CreateAccountHandler
handleRequest(request: IMessage): void
CreateUserHandler
+CreateUserHandler(condition: CreateUserCondition): CreateUserHandler
handleRequest(request: IMessage): void
DefaultHandler
+DefaultHandler(condition: DefaultCondition): DefaultHandler
handleRequest(request: IMessage): void
ExitHandler
+ExitHandler(condition: ExitCondition): ExitHandler
handleRequest(request: IMessage): void
LoginHandler
+LoginHandler(condition: LoginCondition): LoginHandler
handleRequest(request: IMessage): void
MessengerHandler
+MessengerHandler(condition: MessengerCondition): MessengerHandler
handleRequest(request: IMessage): void
PrinterHandler
+PrinterHandler(condition: PrinterCondition): PrinterHandler
handleRequest(request: IMessage): void
ShowBalanceHandler
+ShowBalanceHandler(condition: ShowBalanceCondition): ShowBalanceHandler
handleRequest(request: IMessage): void
StartConversationHandler
+StartConversationHandler(condition: StartConversationCondition): StartConversationHandler
handleRequest(request: IMessage): void
TransactionHandler
+TransactionHandler(condition: TransactionCondition): TransactionHandler
handleRequest(request: IMessage): void
AddExpenseCategoryCondition
+ConditionIsMet(request: IMessage): bool
ChangeSavingsGoalCondition
+ConditionIsMet(request: IMessage): bool
CommandsCondition
+ConditionIsMet(request: IMessage): bool
ConvertionCondition
+ConditionIsMet(request: IMessage): bool
CreateAccountCondition
+ConditionIsMet(request: IMessage): bool
CreateUserCondition
+ConditionIsMet(request: IMessage): bool
DefaultCondition
+ConditionIsMet(request: IMessage): bool
ExitCondition
+ConditionIsMet(request: IMessage): bool
LoginCondition
+ConditionIsMet(request: IMessage): bool
MessengerCondition
+ConditionIsMet(request: IMessage): bool
PrinterCondition
+ConditionIsMet(request: IMessage): bool
ShowBalanceCondition
+ConditionIsMet(request: IMessage): bool
StartConversationCondition
+ConditionIsMet(request: IMessage): bool
TransactionCondition
+ConditionIsMet(request: IMessage): bool
AbstractHandler
- condition: ICondition<IMessage>
handleRequest(request: IMessage): void
AbstractHandler(condition: ICondition<IMessage>): void
+Handler(request: IMessage): void
+AddSuccessor(handler: AbstractHandler<T>): void
«interface»
ICondition
ConditionIsMet(request: IMessage): bool
«interface»
IMessage
GetUserID(): string
SetUserID(userID: string): void
GetMessageText(): string
SetMessageText(messageText: string): void
BotMessage
+GetUserID(): string
+SetUserID(userID: string): void
+GetMessageText(): string
+SetMessageText(messageText: string): void
+BotMessage(ch: string, message: string): BotMessage
«interface»
IPrinter
Print(list: List<Transaction>, path: string)
HTMLPrinter
+HTMLPrinter(): HTMLPrinter
-HeaderStr: List<Transaction>: HeaderRow
-RowsStr: List<Transaction>: ListRows
+Print(list: List<Transaction>, fileName: string): string
ConsoleBot
-instance: ConsoleBot
+GetInstance(): ConsoleBot
-ConsoleBot(): ConsoleBot
+StartCommunication(): void
+SendMessage(id: string, message: string): void
+SendPrint(id: string, path: string): void
AbstractBot
- Bot: ITelegramBotClient
- token: string
- instance: TelegramBot
+GetInstance(): TelegramBot
-TelegramBot(): TelegramBot
+GetClient(): ITelegramBotClient
+OnMessage(sender: object, messageEventArgs: MessageEventArgs)
+StartCommunication(): void
+SendMessage(id: string, message: string): void
+SendPrint(id: string, path: string): void
AbstractBot
-Handler: AbstractHandler<IMessage>
AbstractBot(): AbstractBot
+StartCommunication(): void
+SetChannel(id: string, channel: ICommunicationChannel): void
+SendMessage(id: string, message: string): void
+SendPrint(id: string, path: string): void
+HandleMessage(message: IMessage): void
«interface»
ICommunicationChannel
StartCommunication(): void
SendMessage(id: string, message: string): void
SendPrint(id: string, path: string): void
HandleMessage(message: IMessage): void
Configuration
+StartCommunication(): void
+HandlerSetup(): AbstractHandler<IMessage>
Commands
-instance: Commands
+GetCommandsList(): List<string>
+AddCommand(command: CommandsList: List<string>): void
-Commands(): Commands
+CommandsExists():bool
+HasNoAccountsCommandsList(): List<string>
+HasAccountsCommandsList(): List<string>
+UnloggedCommandsList(): List<string>
+CommandsLabel(id: string): string
Session
+UserInfoMap: Dictionary<string, UserInfo>
-instance: Session
+GetAllUsers(): List<EndUser>
+SetAllUsers(allUsers: List<EndUser>): void
+GetPrinter(): IPrinter
+SetPrinter(printer: IPrinter): void
+GetInstance(): Session
-Session(): Session
+AddUser(username: string, password: string): void
+GetUser(id: string): EndUser
+GetUser(username: string, password: string): EndUser
+SetCommunicationChannel(id: string, newChannel: ICommunicationChannel): void
+GetChatInfo(id:string): UserInfo
UserInfo
+GetConversationState(): ConversationState
+SetConversationState(conversationState: ConversationState): void
+GetUser():EndUser
+SetUserUser: EndUser): void
+GetProvisionalInfo(): Dictionary<string, object>
+SetProvisionalInfo(provisionalInfo: Dictionary<string, object>): void
+UserInfo(): UserInfo
+ClearOperation():void
+GetDictionaryValue(key: string): T
EndUser
+EndUser(username: string, password: string): User
+GetUsername():string
+SetUsername(username: string): void
+GetPassword():string
+SetPassword(password: string): void
+GetExpenseCategories(): List<string>
+SetExpenseCategories(expenseCategories: List<string>): void
+GetAccounts(): ILists<Account>
+ChangeUsername(newUsername: string): void
+ChangePassword(newPassword: string): void
+DisplayAccounts(): string
+AddAccount(name: string, currencyType: Currency, amount: double, maxGoal: SavingsGoal, minGoal
+RemoveAccount(): void
+DisplayExpenseCategories():string
+AddExpenseCategory(newCategory: string):void
+RemoveExpenseCategory(categoryToRemove: string):void
+ContainsExpenseCategory(expenseCategory: string):bool
Account
- currencyExchanger(): CurrencyExchanger
+Account(name: string, currencyType: Currency): void
+GetName(): string
+SetName(name: string): void
+GetCurrencyType(): Currency
-SetCurrencyType(currencyType: Currency): void
+GetAmount(): double
-SetAmount(amount: double): void
+GetType():
+SetType(type: enum): void
+SetSavingsGoal(): SavingsGoal
+SetSavingsGoal(newMax: double, newMin: double): void
-SetTransactionsRecord(): ILists<Transaction>
-SetTransactionsRecord(transactionsRecord: List<Transactions>): void
+Transfer(currency: Currency, amount: double, description: string): void
+ChangeCurrencyType(newCurrencyType: Currency): void
+DisplayAccountType(): string
+ChangeSavingsGoal(newMax: double, newMin: double): void
SavingsGoal
+SavingsGoal(max: double, min: double): SavingsGoal
+GetMax(): double
+SetMax(max: double): void
-GetMin(): double
-SetMin(min: double): void
+GetTimeLimit():DateTime
+SetTimeLimit(timeLimit: DateTime): void
CurrencyExchanger
- instance: CurrencyExchanger
+GetInstance(): CurrencyExchanger
+SetCurrency(List): ILists<Currency>
+SetCurrency(List): ILists<Currency>
+GetCurrency(code: string): CurrencyExchanger
+AddCurrency(type: string, rate: double):void
+RemoveCurrency(type: string):void
+Convert(amount: double, initialCurrency: Currency, finalCurrency: Currency): double
+ExistsCurrency(type: string): bool
Transaction
+GetAmount(): double
+SetAmount(amount: double): void
+GetDescription(): string
+SetDescription(description: string): void
+GetCurrency(): Currency
+SetCurrency(currency: Currency): void
+GetDate():DateTime
+SetDate(date: DateTime): void
+Transaction(amount: double, description: string, currency: Currency, date:DateTime): Transaction
Currency
+Currency(code: string, type: string, rate: double): Currency
+GetType(): string
+SetType(type: string): void
+GetCode():string
+SetCode(code: string): void
+GetExchangeRate(): double
+SetExchangeRate(rate: double): void
«interface»
IAlert
SendAlert(account: Account):string
MaxSavingsGoalAlert
+SendAlert(account: Account):string
MaxSavingsGoalReachedAlert
+SendAlert(account: Account):string
MinSavingsGoalAlert
+SendAlert(account: Account):string
MinSavingsGoalReachedAlert
+SendAlert(account: Account):string