В рамках данного задания необходимо познакомиться с архитектурои˘современных enterprise (и не только) приложении˘, а также практически научиться выделять основные уровни/звенья, на которых оно базируется на примере некоторои˘информационнои˘системы (ИС).

1. Какие основные модели приложении˘Вам известны? Приведите примеры каждого из известных случаев.
2. Объясните разницу между ними, в каких случаях стоит применять тот или инои˘подход?
3. Как Вы понимаете, что скрывается за понятиями тонкого и толстого клиента. Приведите пример яркого представителя thin-клиента.
4. Какая модель является наиболее популярнои˘на текущии˘момент в мире и отлично подходит для большинства решении˘уровня enterprise?

---

1. По моделям приложений, опишу те модели с которыми я сталкивался во временном порядке, по мере ознакомления:
   - Базовая модель это монолитные приложения, познакомился с ними еще на заре процессоров Intel(286, 386) под операционной системой MS-DOS года 90-е прошлого века. Но данная модель до сих пор используется, как пример это офисные приложения или многочисленные редакторы и однопользовательские приложения и etc…
   - Следующая модель с которой столкнулся в конце 90-х это файл-серверная модель. Наиболее популярными файл серверными системами на тот момент мыли системы NovellNetWarea одно из популярных приложений использующих эту модель, на тот момент времени в РФ, было 1С-Бухгалтерия.
   - Плотно получилось поработать (в основном внедрение, конец 90-х начало 2000-х) с клиент-серверной моделью. В этом качестве стоит рассказать про специализированные геоинформационные системы, в данных системах мета, гео и справочная информация хранилась реляционной базе данных (использовались RDBMS: Interbase или Oracle) а растровые и векторные карты записывались и обновлялись на толстых клиентах.
   - На текущий момент большинство решений которые приходиться поддерживать строятся на основе 3-й уровневой и более архитектуре. В основном это 4-х или 5-ти уровневая архитектура состоящая из:
     1. Тонкого клиента или мобильного приложения.
     2. Как правило HTTP-северов с web-интерфейсом или JSON-API.
     3. Серверов бизнес логики(Application servers).
     4. Серверов обмена сообщениями (Middleware servers).
     5. Серверов Баз Данных (Database servers).
   - «Модное» направление - микросервисная архитектура. Опыта пока не имею. Ознакомился с базовыми концепциями на семинаре RedHat Apache Camel.
2. Каждый из этих моделей хорош под свои задачи. Как и упоминалось монолитные приложения до сих пор используются для офисных нужд, хотя и в этой области есть серьезные инновации например google-office. Клиент серверные приложения так же достаточно распространены, например 1С:Предприятие 8.
3. Система как правило с текстовым или графическим интерфейсом для взаимодействия пользователя с (как правило) многоуровневым приложением. Примером тонкого клиента может служить компьютер с браузером, использующийся для работы с веб-приложениями.
4. В текущий момент наиболее популярной на больших и средних предприятиях является многоуровневая модель, благодаря таким преимуществам как масштабируемость, как вертикально так и горизонтально, гибкость и модульность. Например многие компоненты при грамотном планировании и архитектурном решении можно заменять «прозрачно» для клиента.

Следующая часть задания включает практическое выделение слоев отдельно взятоиˇИС: Представим ситуацию: имеется в распоряжении организация «Рога и Копыта», сотрудники которойˇимеют доступ к некоторойˇвнутреннейˇИС. В этойˇсистеме хранится информация о всех сотрудниках организации, обратиться к которойˇможно согласно предоставленному уровню доступа (ролевое разграничение доступа) по следующим правилам:

1. Каждыйˇзарегистрированныйˇсотрудник (имеющийˇлогин/пароль для входа в систему) может получить информацию о любом другом сотруднике: ФИО, название подразделения и город, в котором он/она работает, должность, внутреннийˇномер телефона сотрудника, а также личныйˇпочтовыйˇящик
2. Бухгалтера имеют доступ к заработнойˇплате каждого сотрудника с возможностью ее индексации (увеличения) и депремирования
3. HR-специалисты способны редактировать персональную информацию о каждом, кроме оклада, а также удалять сотрудников из реестра
4. Директор компании (представлен в системе в единственном экземпляре) имеет все права, описанные выше.

От Вас требуется обозначить основные уровни приложения, которые важны во время ее разработки, с кратким описанием предназначения каждого из них. Попытаиˇтесь ответить на вопрос: «Почему этот слойˇвообще представлен и есть ли возможность его не использовать?». Решение даннойˇзадачи позволит в последующих заданиях более детально сконцентрироваться на каждом из перечисленных уровнейˇв отдельности.

---

1. Воспользовавшись многоуровневой моделью выберем 3-х уровневую модель:
   a. Тонкий клиент (web-браузер для сотрудников) или мобильное приложение для.
   b. Серверов (кластера серверов) бизнес логики (Application servers).
   c. Серверов (кластера серверов) Баз Данных (Database servers).
2. Тонкий клиент взаимодействует через необходимый API с требуемыми сервисами.
3. Сообразно поставленной задаче логично выделить следующие модули:
   a. Реестр сотрудников.
   b. Модуль бухгалтерии.
   c. Каталог подразделений.
   d. Редактор Каталога подразделений и реестра сотрудников.
4. На уровне сервисов сосредоточена бóльшая часть бизнес-логики, так же этот уровень проектируются таким образом, чтобы добавление к ним дополнительных экземпляров обеспечивало простое масштабирование производительности серверов приложений.
5. Следующий уровень это слой данных, обеспечивает хранение данных, реализуется средствами СУБД и специализированными каталогами например LDAP, подключение к этому компоненту обеспечивается только с уровня сервера приложений.