Додаток А

Графічний матеріал атестаційної роботи

ГЮИК.501500.006 ПЗ

(позначення документу)

Міністерство освіти і науки України
Харківський Національний Університет Радіоелектроніки

«ЗАТВЕРДЖУЮ»
керівник атестаційної роботи
проф. Іванов В.Г.

Дослідження особливостей розробки сервісів орієнтованих на підтримку динамічного розвитку бізнесу

Графічний матеріал

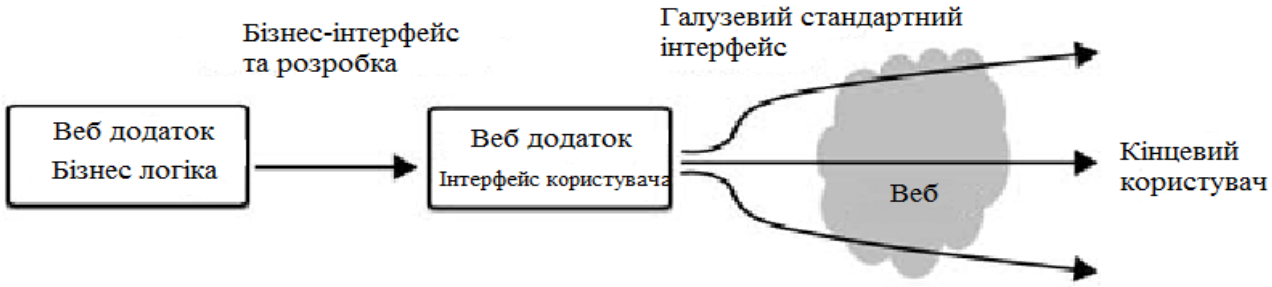
ЛИСТ ЗАТВЕРДЖЕННЯ

ГЮИК.501500.006 – ПЗ

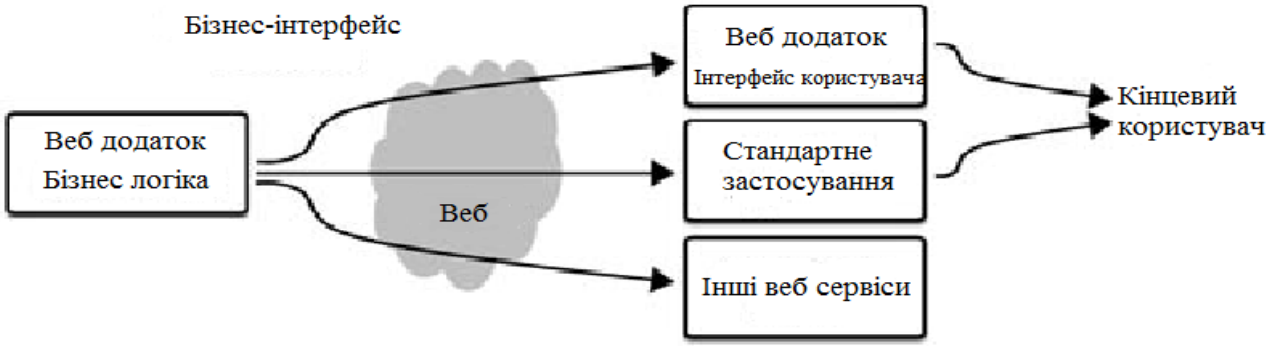
РОЗРОБИВ:
ст. гр. ІТПм-19-1
Стрименешенко О.С.

2020р.

ЕВОЛЮЦІЯ ВЕБ-ДОДАТКІВ ДО ВЕБ-СЕРВІСІВ ТА КЛЮЧОВІ АРХІТЕКТУРНІ ВІДМІННОСТІ



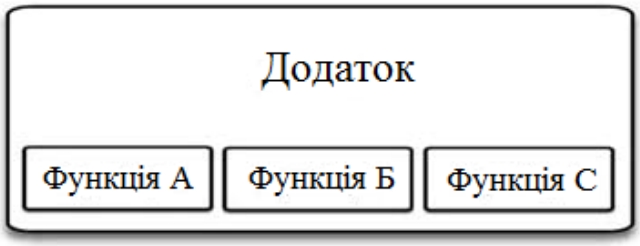
(а) Архітектура веб додатку



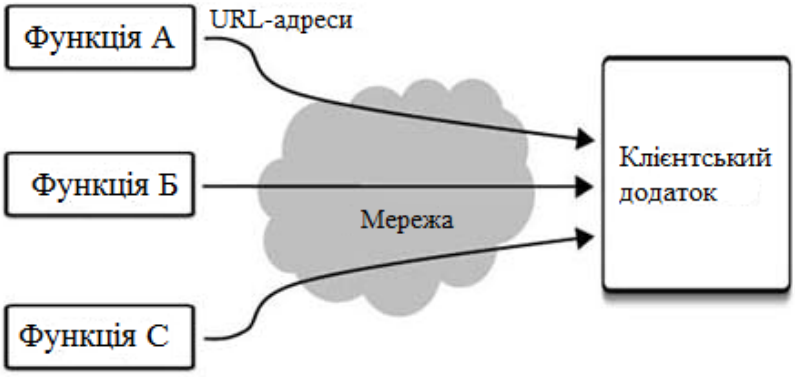
(б) Архітектура веб сервісів

Розроб.	*Стрименешенко О.С.*			*Дослідження особливостей розробки сервісів орієнтованих на підтримку динамічного розвитку бізнесу*	
Перевір.	*Іванов В.Г.*				
Н. Контр.	*Іванов В.Г.*				
				ІТПм-19-1	*Лист 1*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 7*

АРХІТЕКТУРНІ ВІДМІННОСТІ МІЖ МОНОЛІТНИМ ДОДАТКОМ ТА РОЗПОДІЛЕНИМ ДОДАТКОМ



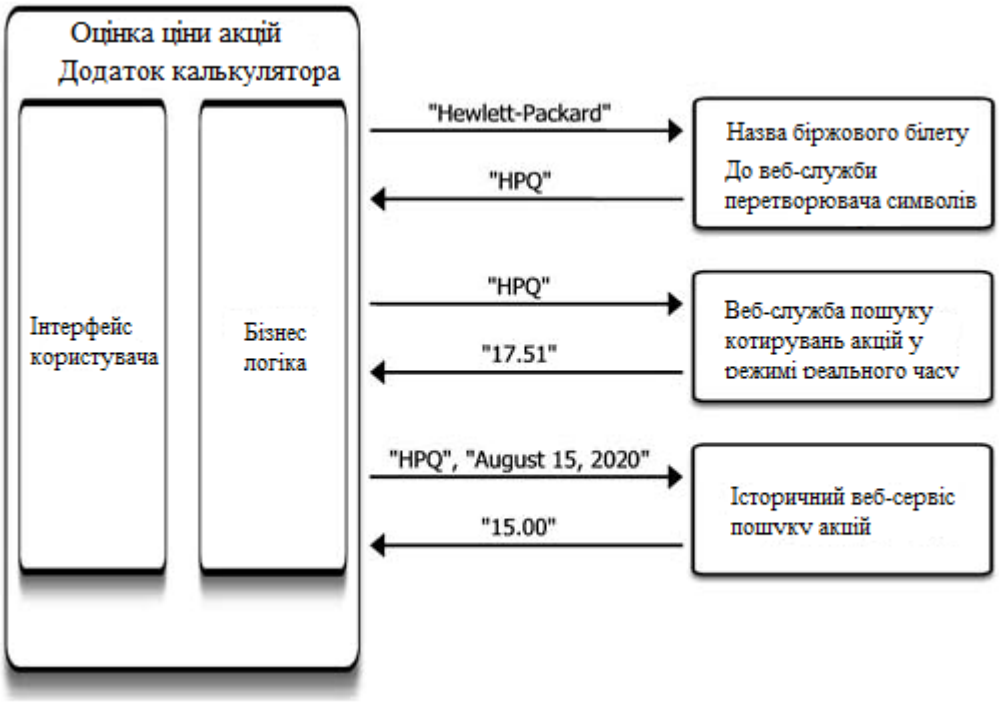
(а) Монолітний додаток с інтегрованими функціями



(б) Клієнтський додаток, що викликає віддалені функції веб-служб

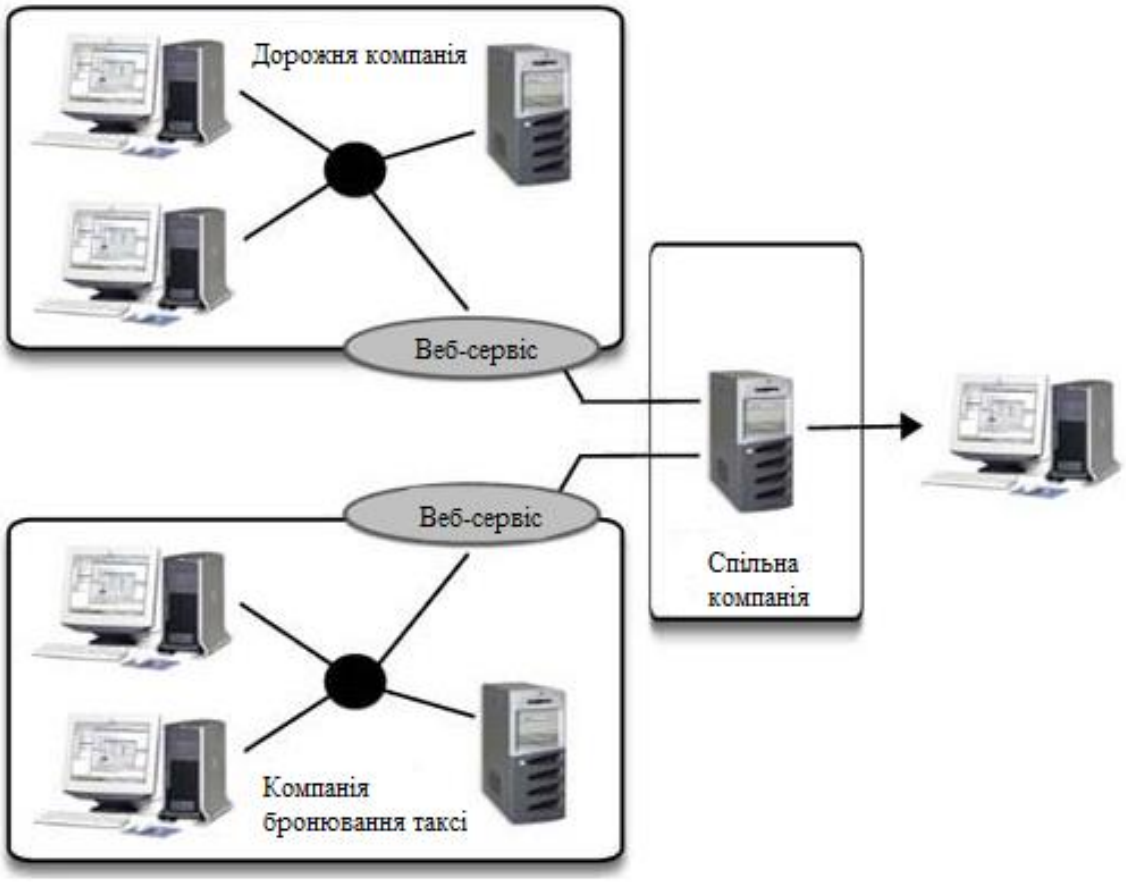
Розроб.	*Стрименешенко О.С.*			*Дослідження особливостей розробки сервісів орієнтованих на підтримку динамічного розвитку бізнесу*	
Перевір.	*Іванов В.Г.*				
Н. Контр.	*Іванов В.Г.*				
				ІТПм-19-1	*Лист 2*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 7*

НАДСИЛАННЯ ТА ОТРИМАННЯ ПОВІДОМЛЕНЬ ВЕБ-СЕРВІСУ



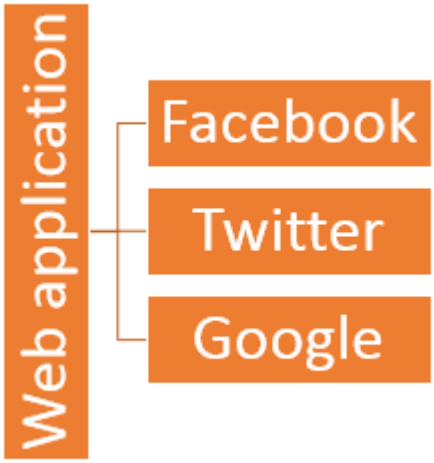
Розроб.	*Стрименешенко О.С.*			*Дослідження особливостей розробки сервісів орієнтованих на підтримку динамічного розвитку бізнесу*	
Перевір.	*Іванов В.Г.*				
Н. Контр.	*Іванов В.Г.*				
				ІТПм-19-1	*Лист 3*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 7*

СКЛАДАЮЧИ РАЗОМ ПОСЛУГИ, ЩО ВИСТАВЛЯЮТЬСЯ БАГАТЬМА КОРПОРАЦІЯМИ



Розроб.	*Стрименешенко О.С.*			*Дослідження особливостей розробки сервісів орієнтованих на підтримку динамічного розвитку бізнесу*	
Перевір.	*Іванов В.Г.*				
Н. Контр.	*Іванов В.Г.*				
				ІТПм-19-1	*Лист 4*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 7*

ПРИКЛАД ВЕБ-ПРОГРАМИ, ЯКА СПІЛКУЄТЬСЯ З ІНШИМИ

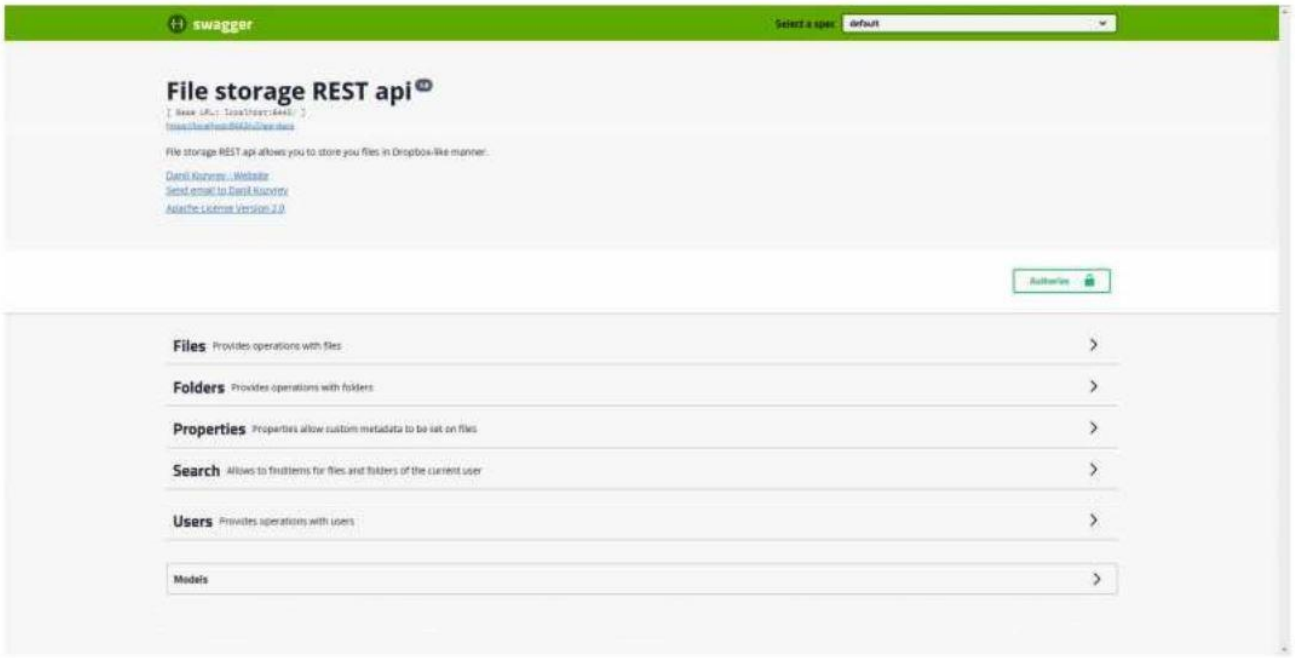


Розроб.	*Стрименешенко О.С.*			*Дослідження особливостей розробки сервісів орієнтованих на підтримку динамічного розвитку бізнесу*	
Перевір.	*Іванов В.Г.*				
Н. Контр.	*Іванов В.Г.*				
				ІТПм-19-1	*Лист 5*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 7*

СТРУКТУРА JWT

Розроб.	*Стрименешенко О.С.*			*Дослідження особливостей розробки сервісів орієнтованих на підтримку динамічного розвитку бізнесу*	
Перевір.	*Іванов В.Г.*				
Н. Контр.	*Іванов В.Г.*				
				ІТПм-19-1	*Лист 6*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 7*

СПИСОК РЕСУРСІВ



Розроб.	*Стрименешенко О.С.*			*Дослідження особливостей розробки сервісів орієнтованих на підтримку динамічного розвитку бізнесу*	
Перевір.	*Іванов В.Г.*				
Н. Контр.	*Іванов В.Г.*				
				ІТПм-19-1	*Лист 7*
Затверд.	*Гребеннік І.В.*			*СТ*	*Листів 7*

Додаток Б

Текст програми

ГЮИК.501500.006 – ПЗ

Міністерство освіти і науки України
Харківський Національний Університет Радіоелектроніки

«ЗАТВЕРДЖУЮ»
керівник атестаційної роботи
проф. Іванов В.Г.

Дослідження особливостей розробки сервісів орієнтованих на підтримку динамічного розвитку бізнесу

Текст програми

ЛИСТ ЗАТВЕРДЖЕННЯ

ГЮИК.501500.006 – ПЗ

РОЗРОБИВ:
ст. гр. ІТПм-19-1
Стрименешенко О.С.

2020р.

```
@Api(tags = "auth",
              consumes = MediaType.APPLICATION_JSON_VALUE,
              produces = MediaType.APPLICATION_JSON_VALUE)
@RequestMapping(value = "/api/auth", headers = SwaggerConfiguration.HEADER_VERSION)
@RestController
@Lazy
public class AuthenticationRestController extends BaseSpringMvcController {

       /**
        * access token ttl = 24 hours
        */
       public static final int TIME_BEFORE_EXPIRE = 1440;

       private final AuthTokenService authTokenService;
       private final AuthorizationValidator validator;
       private final LoginStatisticsService loginStatisticsService;
       private final RefreshTokenValidator refreshTokenValidator;

       @Lazy
       public AuthenticationRestController(AuthTokenService authTokenService,
AuthorizationValidator validator, LoginStatisticsService loginStatisticsService,
RefreshTokenValidator refreshTokenValidator) {
              this.authTokenService = authTokenService;
              this.validator = validator;
              this.loginStatisticsService = loginStatisticsService;
              this.refreshTokenValidator = refreshTokenValidator;
       }

       @ApiOperation(value = "Authorize",
                     nickname = "authorize",
                     tags = "auth",
                     response = AuthorizeResponse.class)
       @PostMapping("/authorize")
       @HeaderPreferredLocale
       public ResponseEntity<AuthorizeResponse> authorize(HttpServletRequest request,
@RequestBody AuthorizeRequest authorizeRequest) {
              ErrorManager errorManager = new ErrorManager();
              return super.handleException(cryptoBundle -> {
                     AuthorizeResponse response = new AuthorizeResponse();
                     validator.validate(new BaseFormValidationContext<>(authorizeRequest,
errorManager));
                     if (errorManager.isErrorsPresent()) {
                            response.setErrorManager(errorManager);
                     } else {
                            LoginBean loginBean = LoginManager.get();

       loginBean.setLoginInput(authorizeRequest.getUsername().getValue());
                            loginBean.setPswd(authorizeRequest.getPassword().getValue());
                            loginBean.checkLogin();
                            ErrorManager localErrorManager = loginBean.getLoginResult();
                            LoginStatistik loginStatistik =
loginStatisticsService.createDefaultLoginStatistik(loginBean, request.getRemoteAddr());
                            if (localErrorManager.isErrorsEmpty()) {

       response.setAuthToken(authTokenService.authenticate(loginBean.getLoginInput(),
loginBean.resolveBenutzerVornameNachname(), false, TIME_BEFORE_EXPIRE));

       loginStatisticsService.saveSuccessfullLoginStatistik(loginStatistik, loginBean);
                            } else {
                                   errorManager.appendErrorsAndHinweises(localErrorManager);
                                   loginStatisticsService.saveLoginStatistik(loginStatistik);
                                   response.setErrorManager(errorManager);
                            }
                     }
                     return response;
```

```
                }, errorManager);
        }

        @ApiOperation(
                        value = "Refresh token",
                        nickname = "refreshToken",
                        tags = "auth",
                        response = AuthorizeResponse.class)
        @PostMapping("/refresh-token")
        @HeaderPreferredLocale
        public ResponseEntity<AuthorizeResponse> refreshToken(@RequestBody
RefreshTokenRequest request) {
                ErrorManager errorManager = new ErrorManager();
                return super.handleException(cryptoBundle -> {
                        AuthorizeResponse response = new AuthorizeResponse();
                        refreshTokenValidator.validate(new BaseFormValidationContext<>(request,
errorManager));
                        if (errorManager.isErrorsPresent()) {
                                response.setErrorManager(errorManager);
                        } else {

        response.setAuthToken(authTokenService.refreshToken(request.getRefreshToken().getValu
e(), request.getSignature().getValue()));
                        }
                        return response;
                }, errorManager);
        }

}
```

```
public class ContentElementUploadFileUrlControllerAction extends
AbstractContentElementUrlControllerAction {

	@Autowired
	private FileDataService fileDataService;

	@Override
	public int execute(HttpServletRequest request, HttpServletResponse response) throws
IOException, JSONException {
		ContentElementFileUploadForm fileUploadForm = new
ContentElementFileUploadForm();

		CsrfGuard.getInstance().restorePrevToken(request);
		PostResult postResult = PostUtil.doPostMultipart(request, fileUploadForm);

		boolean shouldFileBeSavedAsNew =
fileUploadForm.getContentFormExternalFile2Upload().getInputBytes().length > 0;

		if ((shouldFileBeSavedAsNew &&
fileUploadForm.getContentFormExternalFile2Upload().getSize().getInt() >
StandardConstants.getMaxUploadFileSizeBit()) || postResult.getErrors().containsKey("size"))
{
			JSONObject result = new JSONObject();
			result.put("errorMessage",
StandardErrors.DOCUMENT_UPLOAD_FILESIZE_ERROR_TEXT);

			response.getWriter().println(result.toString());
			return TagSupport.SKIP_PAGE;
		}

		if (!shouldFileBeSavedAsNew && !new NotNegativeAllowedRange<Long>(new Long(1),
AllowedRange.MIN,
0l).validate(fileUploadForm.getContentFormExternalFile2Upload().getFileId(), "fileId",
"Datei", Genus.FEMININUM, Measure.KEIN, new HashMap<String, String>())) {
			JSONObject result = new JSONObject();
			result.put("errorMessage", "Bitte wählen Sie eine Datei aus.");

			response.getWriter().println(result.toString());

			return TagSupport.SKIP_PAGE;
		}
		FileDataInfo fileDataInfo = shouldFileBeSavedAsNew ? new FileDataInfo() :
fileDataService.getById(fileUploadForm.getContentFormExternalFile2Upload().getFileId().getNa
tivevalue());
		if (shouldFileBeSavedAsNew) {

	fileDataInfo.setFilename(fileUploadForm.getContentFormExternalFile2Upload().getFilena
me().getValue());

	fileDataInfo.setSize(fileUploadForm.getContentFormExternalFile2Upload().getSize().get
Int());

	fileDataInfo.setMimetype(fileUploadForm.getContentFormExternalFile2Upload().getMimety
pe().getValue());
			fileDataInfo.setOwnerType(PortalOwnerType.FAQINFO);

	fileDataInfo.setOwnerId(LoginManager.get().doGetBenutzer().getPkbenutzerid());
		}

	fileDataInfo.setDescription(fileUploadForm.getContentFormExternalFile2Upload().getDes
cription().getValue());
		fileDataService.save(fileDataInfo);

		if (shouldFileBeSavedAsNew) {
```

```
                    fileDataService.saveResourceOnFileSystem(fileDataInfo,
fileUploadForm.getContentFormExternalFile2Upload().getInputBytes());
                }

                JSONObject result = getJSONObject4ExternalFile(fileDataInfo);
                response.getWriter().println(result.toString());

                return TagSupport.SKIP_PAGE;
        }

}
```

Додаток В

Відомість дипломної роботи

№	Позначення	Найменування	Додаткові відомості
		Текстові документи	
1	ГЮИК.501500.006 ПЗ	Пояснювальна записка	65 листів
2	ГЮИК.501500.006 ПЗ 34 01 – 01 12 01	Текст програми	4 листів
3	ГЮИК.501500.006 ПЗ	Графічні документи	7 листів
4		Еволюція веб-додатків до веб-	1 лист
		сервісів та ключові архітектурні відмінності	
5		Архітектурні відмінності між	1 лист
		монолітним додатком та розподіленим додатком	
6		Надсилання та отримання повідомлень веб-сервісу	1 лист
7		Складаючи разом послуги, що виставляються багатьма корпораціями	1 лист
8		Приклад веб-програми, яка	1 лист
		спілкується з іншими	
9		Структура jwt	1 лист
10		Список ресурсів	1 лист

					ГЮИК.501500.006 Д3			
Зм.	*Лист*	*№ докум.*	*Підп*	*Дата*				
Розробив		*Стрименешенко О.С.*			*Дослідження особливостей розробки сервісів орієнтованих на підтримку динамічного розвитку бізнесу*		Лист	Листів
Перевірив		*Іванов В.Г.*					*1*	*1*
Н. Контр.		*Іванов В.Г.*				*ХНУРЕ Кафедра СТ*		
Затверд.		*Гребенник І.В.*						